

**LAB # 09**

**The Purpose of this Lab is to get familiar the Tables Joins Which Includes Inner, Outer & Natural Joins**

**Lab Exercise**

**Question No. 1**

**We have the following tables which creates statement are**

1. **create table supplier(S\_NO int identity(1,1), SUP\_ID varchar(50) primary key,SUP\_NAME varchar (50), SUP\_ADD varchar (50), SUP\_NIC int, rank int)**
2. **create table Product(Prod\_ID varchar(50) primary key,Prod\_Name varchar(50),[Price/Unit] float)**
3. **create table Orders(Ord\_ID varchar(50) primary key,Prod\_ID varchar(50),Qty int,Totalrate int,SUP\_ID varchar (50))**

**Write the Queries using Joins on above tables**

1. **List of all supplier that palced order**
2. **List of all Product that are supplied by supplier whose id is 101**
3. **Find all order(s) of product named Rice**

**Code:**

# Q.1

SELECT s.SUP\_ID, s.SUP\_NAME

FROM supplier s INNER JOIN orders o

ON s.SUP\_ID = o.SUP\_ID

# Q.2

SELECT p. Prod\_Name, s.SUP\_ID

FROM Product p INNER JOIN Supplier s

ON p.Prod\_ID = s.SUP\_ID

WHERE s.SUP\_ID =101

# Q.3

SELECT \*

FROM orders o INNER JOIN Product p

ON o.Prod\_ID = p.Prod\_ID

WHERE p.Prod\_Name=’Rice’



**LAB # 10**

**The Purpose of this Lab is to get familiar that how to use Groups & Having Clauses with Aggregates Functions**

**Lab Exercise**

**Question No. 1**

1. **List average salary of each job.**
2. **Find average and sum of all the salaries of each job excluding clerks.**
3. **Find average and sum of the salaries of each job excluding salesmen', clerk' and 'manager'.**
4. **Find count, sum and average salaries of each job excluding salesmen', clerk' and 'manager'.**
5. **List average salary of each department.**

**Code:**

# Q.1

SELECT Job\_ID, AVG (salary)

FROM company

GROUP BY Job\_ID;

SELECT Job\_ID , AVG(salary) ,SUM (salary)

FROM company

WHERE Job\_ID <> ‘clerk’

GROUP BY Job\_ID;

# Q.2

SELECT Job\_ID , AVG(salary) ,SUM (salary)

FROM company

WHERE Job\_ID <> (‘salesmen’,‘clerk’,’manager’)

GROUP BY Job\_ID;

# Q.3 and 4

SELECT Job\_ID ,COUNT(salary), AVG(salary) ,SUM (salary)

FROM company

WHERE Job\_ID <> (‘salesmen’,‘clerk’,’manager’)

GROUP BY Job\_ID;

# Q.5

SELECT Depart\_ID, AVG (salary)

FROM company

GROUP BY Depart\_ID;



**LAB # 11**

**The Purpose of this Lab is to get familiar with the Analytical Queries Which includes Rollup, Cube & Top- N (Limits) Queries**

**Lab Exercise**

**Question No. 1**

1. **Find the names of Top 10 employees which salaries are highest.**

**Code:**

SELECT E\_Names , salary

FROM company

ORDER BY salary DESC

LIMIT 10



**LAB # 14**

**The Purpose of this Lab is to get familiar that how to use SQL Stored Procedures with One Parameter & Multiple Parameters**

**Lab Exercise**

**QuestionNo1**

**Create a stored procedure with parameters and it will insert a new row into the Names table with the FirstName and LastName columns And call stored procedure with parameters.**

**QuestionNo2**

**Create a stored procedure with input parameters as input student registration number, total marks and number of subjects and updates the percentage of marks**

**Explanation:**

* **Names of the procedure are STUDENT\_MARKS**
* **Defines parameter STUDENT\_REG\_NO (character data type of length 15), TOTAL\_MARKS (decimal data type) and NO\_SUBJECTS (integer type) which all are input parameters.**
* **Indicates the procedure is an SQL procedure that modifies SQL data.**
* **Defines the procedure body as a single UPDATE statement. When the procedure is called, the UPDATE statement is executed using the values passed for STUDENT\_REG\_NO, TOTAL\_MARKS, and NO\_SUBJECTS**
* **Defines the procedure body**

**Code:**

# Q. 1

DELIMITER//

CREATE PROCEDURE InsertPersonalDetails (IN cname VARCHAR(50))

BEGIN

SELECT \*

FROM employees;

WHERE firstname=cname && lastname=cname;

END

//DELIMITER;

CALL InsertPersonalDetails ("Muhammad", "Ali");

# Q.2

DELIMITER //

CREATE PROCEDURE STUDENT\_MARKS (IN STUDENT\_REG\_NO CHAR(15),IN TOTAL\_MARKS DECIMAL(7,2), NO\_SUBJECTS INT(3))

BEGIN

UPDATE STUDENT\_MARKS.MARKS;

SET PERCENTAGE = TOTAL\_MARKS/NO\_SUBJECT;

WHERE REG\_NO = STUDENT\_REG\_NO;

END

// DELIMITER;