Syeda Reeha Quasar

14114802719

4C7

Aim

Write the SQL queries to implement the joins.

Experiment - 8

DATABASE MANAGEMENT SYSTEMS LAB

# **EXPERIMENT – 9**

## **Aim:**

Write the SQL queries to implement the joins.

## **Tools Used:**

MariaDB

## **Procedure/ Queries:**

SQL joins are used to fetch / retrieve data from two or more data tables, based on join condition. A join condition is a relationship among some columns in the data tables that take part in SQL join. Basically data tables are related to each other with keys. We use keys relationship in SQL joins.

**Types of Joins:** in SQL Server we have only three types of joints. Using these joins we fetched the data from multiple tables based on condition.

1. **INNER JOIN**: Inner join returns only those records/rows that match/exists in both the tables. Syntax for inner join is as:

Select \* from table\_1 as t1

Inner join table\_2 as t2

On t1.IDcol=t2.IDccol

1. **OUTER JOIN:** We have three types of outer joins:
2. **Left Outer Join:** Left outer join returns all records/rows from left table and from right table returns only matched records. If there are no columns matched in the right table, it returns NULL values. Syntax for Left outer Join is as:

Select \* from table1 as t1

Left outer join table2 as t2

On t1.IDcol=t2.IDcol

1. **Right Outer Join:** Right outer join returns all records/rows from right table and from left table returns only matched records. If there are no columns matched in the right table, it returns NULL values. Syntax for Left outer Join is as:

Select \* from table1 as t1

Right outer join table2 as t2

On t1.IDcol=t2.IDcol

1. **Full outer Join:** full outer join combines left outer join and right outer join. This joint returns all records/rows from both the tables. If there are no columns matching in both the tables, it returns NULL values. Syntax for full outer join is as:

Select \* from table1 as t1

Full outer join table2 as t2

On t1.IDcol=t2.IDcol

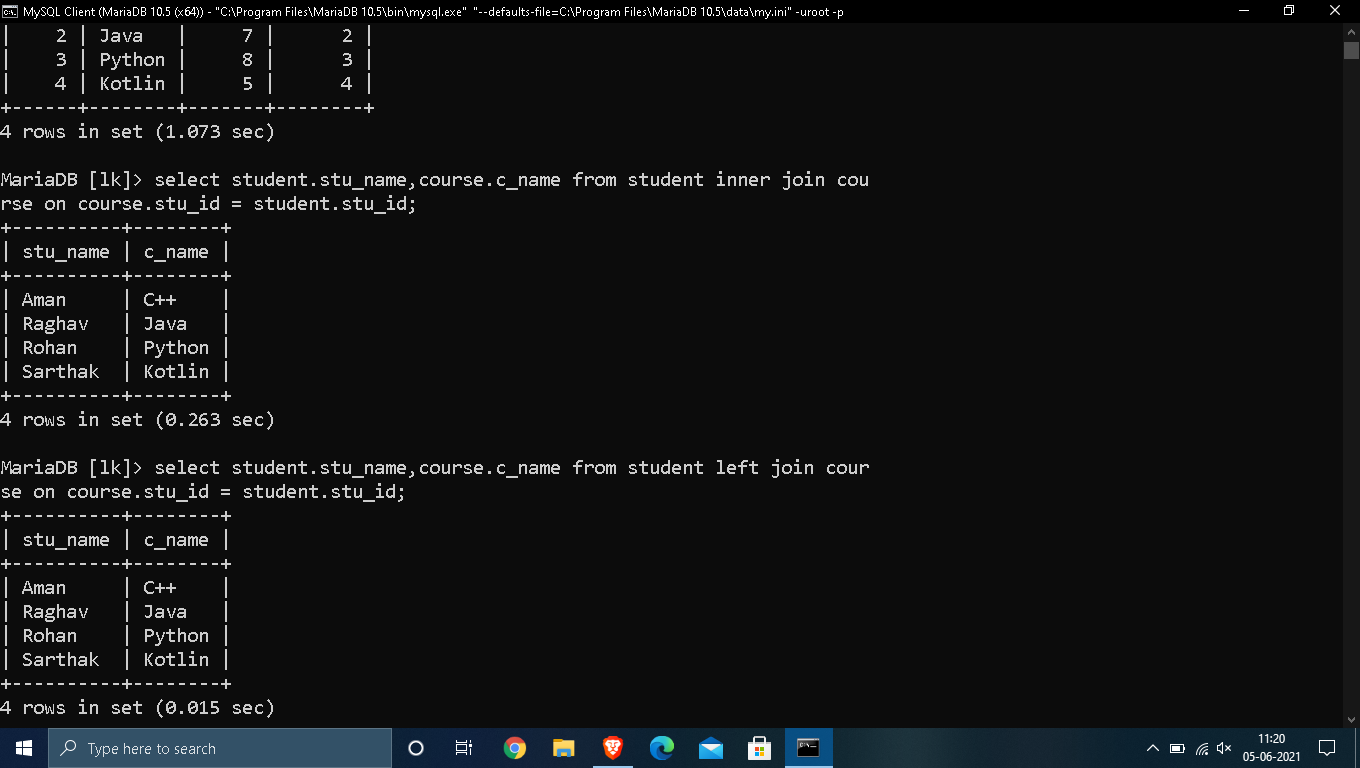
1. **CROSS JOIN:** cross join is the cartesian product of both tables. This joint does not need any condition to join two tables. This joint returns records/rows that are multiplication of records number from both the tables means each row on left table will be related to each row of right table. Syntax for cross join is:

Select\*from table1

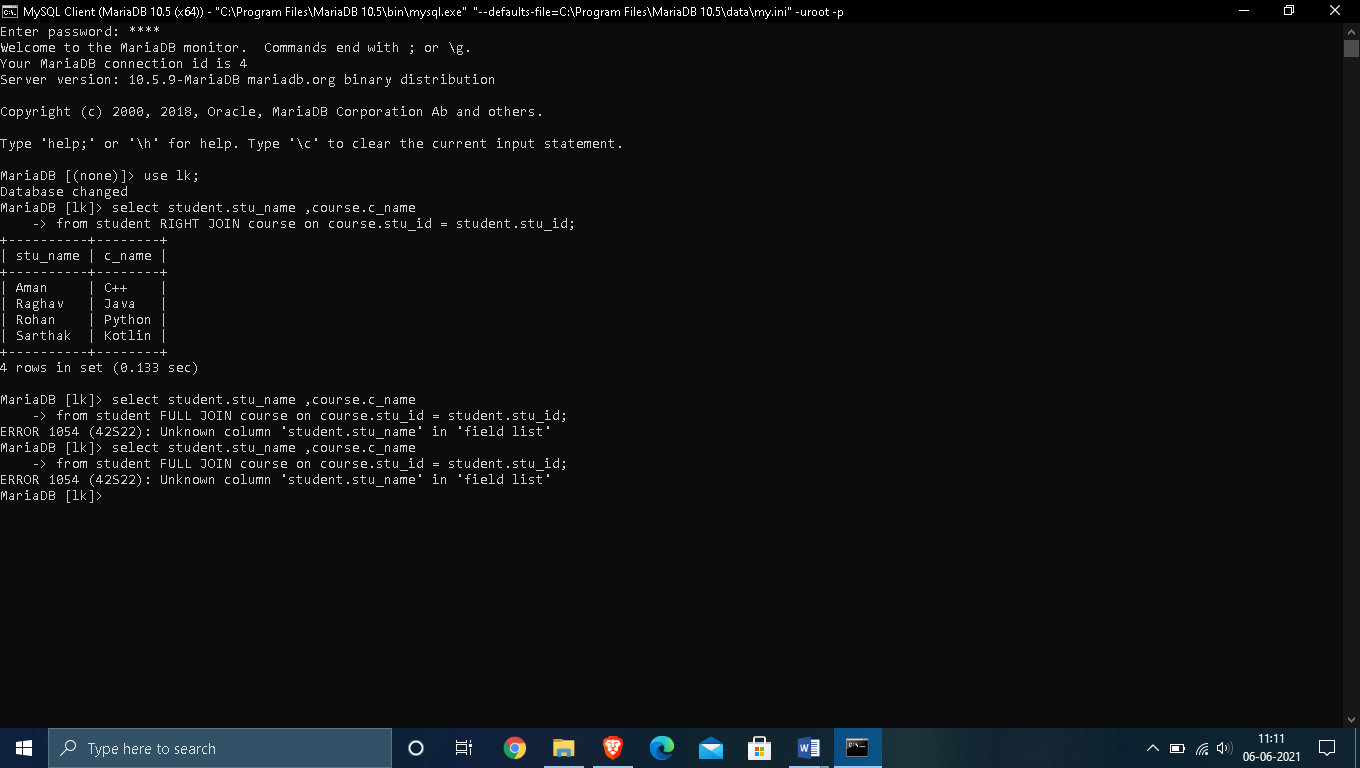
Cross join table2

1. **SELF JOIN:** Self join is used to join a data base table to itself, particularly when the table has a foreign key that references it's own primary key. Basically we have only three types of joins: inner join, outer join and cross join. We use any of these three joints to join a table to itself. Hence self join is not a type of SQL join**.**

## INNER JOIN & LEFT JOIN :



## 2. RIGHT JOIN:



# QUERIES:

## Find out the products which have been sold to ‘Ivan Bayross’.

**Input Query:**

**select** c.**name**,p.production,p. description,s.orderno

**from** client\_master c,product\_master1 p,sales\_order s,sales\_order\_details so

**where** c.**name**="Ivan" **and** c.clientno=s.clientno **and** s.orderno=so.orderno

**and** so.productno=p.production;

**OUTPUT:**

**Graphical user interface, text, application

Description automatically generated**

## Find out the products and their quantities that we have to deliver in the current month.

**Input Query:**

**SELECT** SOD.PRODUCTNO, PM.DESCRIPTION, **SUM**(SOD.QTYORDERED)

**FROM** SALES\_ORDER\_DETAILS SOD, SALES\_ORDER SO, product\_master1 PM

**WHERE** PM.PRODUCTION = SOD.PRODUCTNO **AND** SO.ORDERNO = SOD.ORDERNO

**AND** **monthname**(SO.OrderDate)=**monthname**(**curdate**())

**GROUP** **BY** SOD.PRODUCTNO, PM.DESCRIPTION;

**OUTPUT:**

**Graphical user interface, text, application

Description automatically generated**

## list the product number and the description of the constantly sold(i.e rapidly moving) products.

**Input Query:**

**select** s.productno,p.description

**from** sales\_order\_details s,product\_master1 p

**where** s.productno=p.production

**and** ProductRate=(**select** **max**(ProductRate) **from** sales\_order\_details);

**OUTPUT:**

**Graphical user interface, text, application, email

Description automatically generated**

## find the names of clients who have purchased ‘Trousers’.

**Input Query:**

**select** p.production,p.description,c.**name**

**from** product\_master1 p,client\_master c,sales\_order\_details so,sales\_order s

**where** description="Trousers" **and** p.production=so.productno

**and** c.clientno=s.clientno;

**OUTPUT:**

**Graphical user interface, text, application, email

Description automatically generated**

## list of products and orders from customers who have ordered less than five units of ‘Pullovers’.

**Input Query:**

**select** p.production,p.description,s.orderno

**from** product\_master1 p,sales\_order s,sales\_order\_details so

**where** p.description="Pullovers" **and** p.production=so.productno

**and** s.orderno=so.orderno **and** qtyordered<5;

**OUTPUT:**

**Graphical user interface, text, application

Description automatically generated**

## Find the products and their quantities for the orders placed by ‘Ivan Bayross’ and ‘Mamta Muzumdar’.

**Input Query:**

**select** p.production,p.description,**sum**(qtyordered)"QUANTITY ORDERED"

**from** product\_master1 p,sales\_order s,sales\_order\_details so,client\_master c

**where** s.orderno=so.orderno **and** p.production=so.productno

**and** c.clientno=s.clientno **and**(c.**name**="Ivan" **or** c.**name**="Mamta Mazumdar")

**group** **by** so.productno,p.description;

**OUTPUT:**

**Graphical user interface, text, application

Description automatically generated**

## Find the products and their quantities for the orders placed by client number ‘C00001’ and ‘C00002’.

**Input Query:**

**SELECT** SO.CLIENTNO, SOD.PRODUCTNO, PM.DESCRIPTION, **SUM**(qTYORDERED) UNITSORDERED

**FROM** SALES\_ORDER SO, SALES\_ORDER\_DETAILS SOD, product\_master1 PM, CLIENT\_MASTER CM

**WHERE** SO.ORDERNO = SOD.ORDERNO **AND** SOD.PRODUCTNO = PM.PRODUCTION **AND** SO.CLIENTNO = CM.CLIENTNO

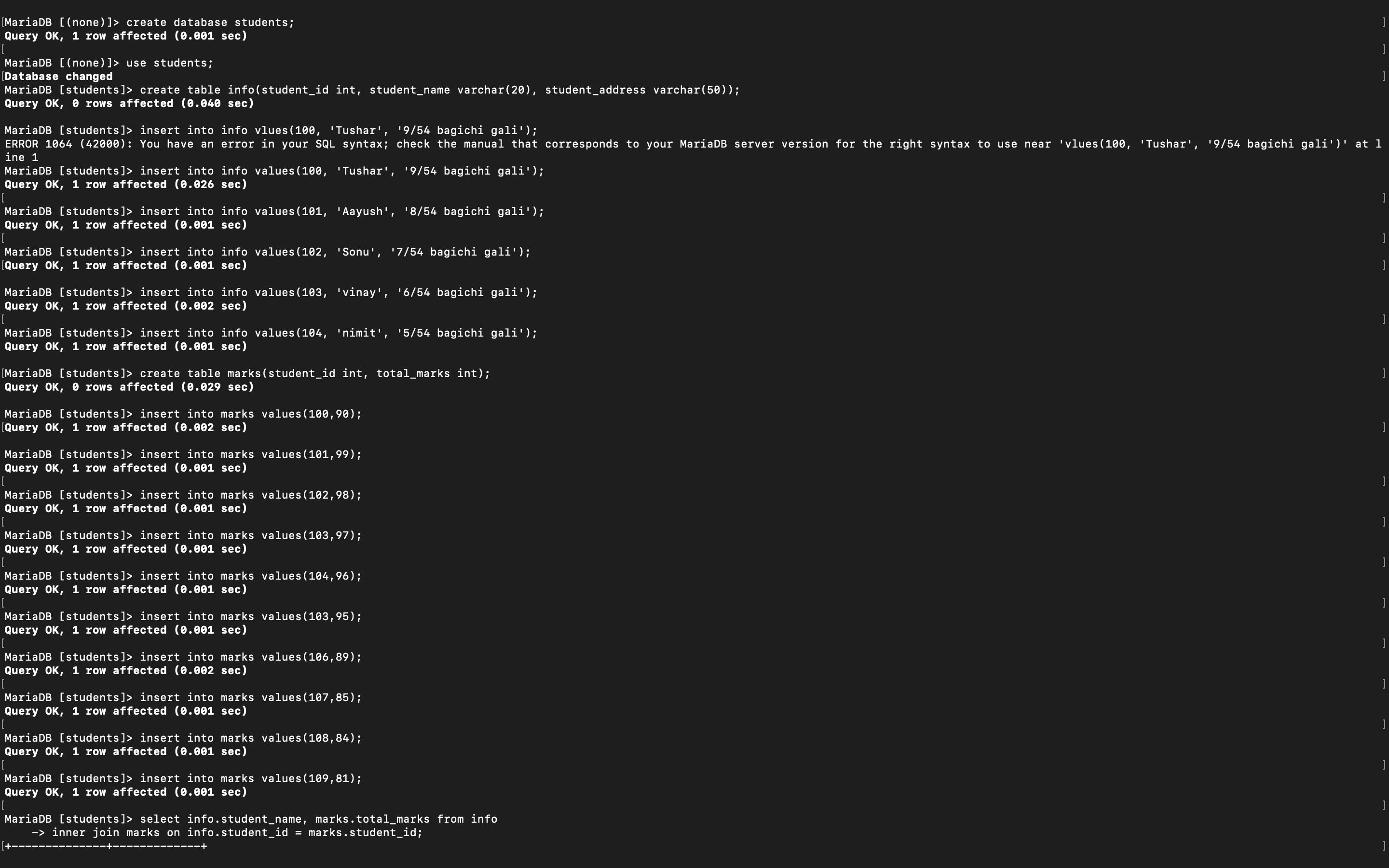
**GROUP** **BY** SO.CLIENTNO, SOD.PRODUCTNO, PM.DESCRIPTION

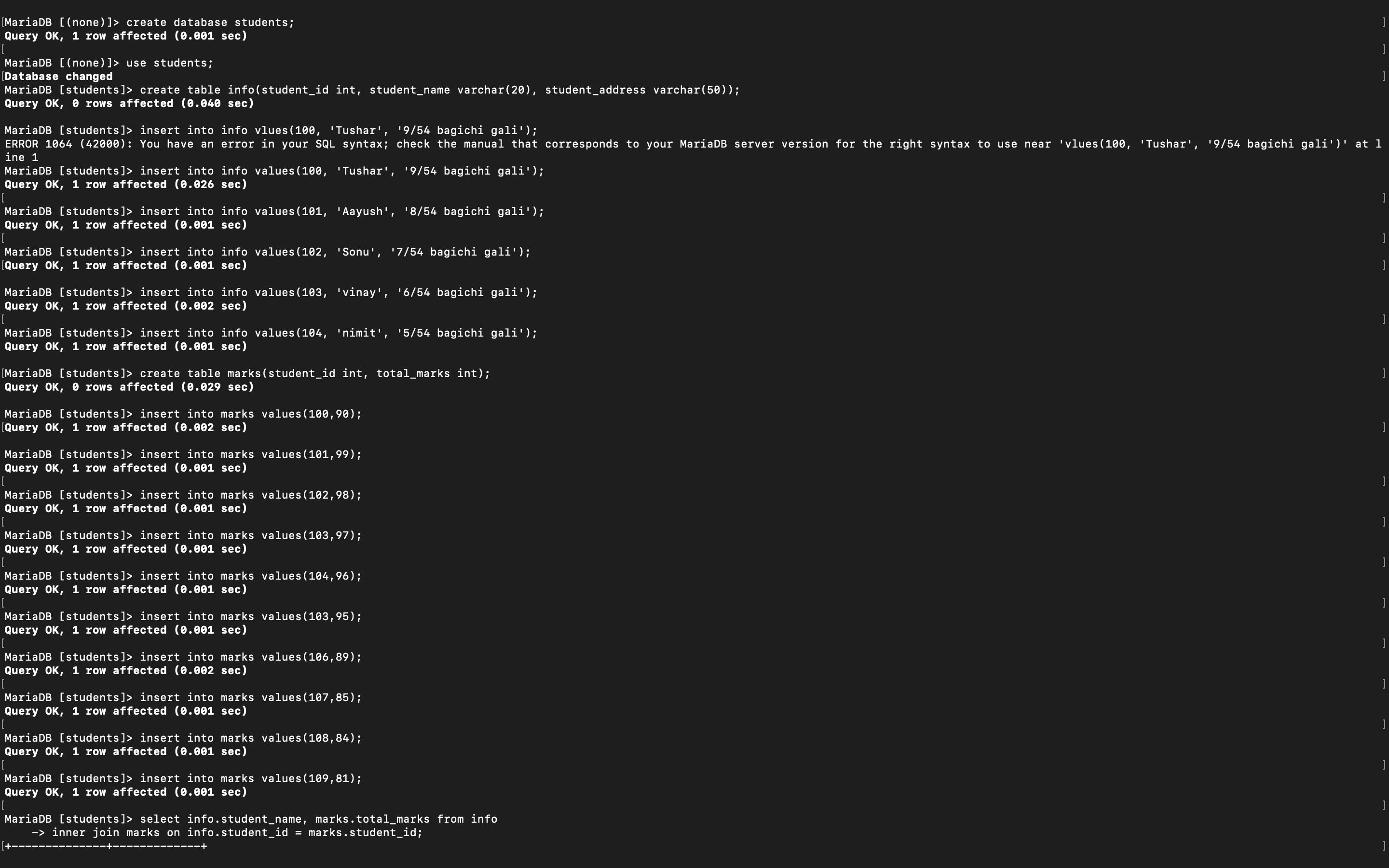
**HAVING** SO.CLIENTNO = 'C00001' **OR** SO.CLIENTNO = 'C00002';

**OUTPUT:**

A picture containing text

Description automatically generated





# VIVA QUESTIONS:

## Q.1: What is a JOIN Operation?

**Ans:**

A join is an SQL operation performed to establish a connection between two or more database tables based on matching columns, thereby creating a relationship between the tables. A Join operation combines related tuples from different relations, if and only if a given join condition is satisfied. It is denoted by ⋈.

## Q.2: What is the difference between JOIN and PRODUCT operations?

**Ans:**

|  |  |  |
| --- | --- | --- |
| **S.no** | **JOIN Operation** | **PRODUCT Operations** |
| 1. | Natural join is denoted by ‘⋈’. It is applicable only when there is atleast one attribute common between 2 relations. | Product operation is denoted by  X \*Y and returns a relation on tuples, whose schema contains all fields of X followed by all fields of Y. |
| 2. | If you will use join, you will have keys which will be able to join rows from first table with rows of second table and depending on your relation, number of rows may vary. | Cartesian product means you will have rows with each record from one table matched with all rows of second table. |
| 3. | JOIN operation is applied only with a WHERE clause. | In product operations, WHERE clause is not required. |

## Q.3: What are different types of JOIN operations?

**Ans:**

There are mainly 3 types of joins in SQL: Inner Join, Outer Join and Cross Join.

* **(INNER) JOIN:** Returns records that have matching values in both tables
* **LEFT (OUTER) JOIN**: Returns all records from the left table, and the matched records from the right table
* **RIGHT (OUTER) JOIN:** Returns all records from the right table, and the matched records from the left table
* **FULL (OUTER) JOIN:** Returns all records when there is a match in either left or right table.
* **CROSS JOIN:** This join returns records/rows that are multiplication of record number from both the tables means each row on the left table will relate to each row of right table.

## Q.4: Difference between RIGHT OUTER JOIN and LEFT OUTER JOIN?

**Ans:**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **RIGHT OUTER JOIN** | **LEFT OUTER JOIN** |
|  | Fetches all the rows from the table on the right,regardless of whether there are any matching columns in the “left” table. | Fetches all the rows from the table on the left, regardless of whether there are any matching columns in the “right” table. |
| 2. | The result of Right Outer Join can be seen as: Inner Join +all the unmatched rows from the right table. | The result of Left Outer Join can be seen as: Inner Join +all the unmatched rows from the left table. |
| 3. | Unmatched data of the left table is lost. | Unmatched data of the right table is lost. |
| 4. | If left table doesn’t have the matching record then for such records left table column will have NULL value in the result. | If right table doesn’t have the matching record then for such records right table column will have NULL value in the result. |

## Q.5: What is an INNER Join?

**Ans:**

Inner Join returns only those records/rows that match/exists in both the tables.Inner Join clause in SQL creates a new table (not physical) by combining rows that have matching values in two or more tables. This join is based on a logical relationship (or a common field) between the tables and is used to retrieve data that appears in both tables. Syntax for Inner Join is as follows:

Select\* from table\_1 as t1

inner join table\_2 as t2

on t1.IDcol=t2.IDcol;