

**NATIONAL INSTITUTE OF TECHNOLOGY SILCHAR**

**Cachar, Assam**

**B.Tech. V<sup>th</sup> Sem**

**Subject Code:** CS-312

**Subject Name:** Database Management System

**Submitted By:**

Name : Subhojit Ghimire

Sch. Id. : 1912160

Branch : CSE – B

CONNECTION: 1912160\_CS312

CREATE SCHEMA `assignment5`;

### 1. i. student

Result Grid				
	S_ID	Name	Address	Age
	1	LOKI	DELHI	19
	2	KISHAN	KERELA	20
	3	RISHI	ASSAM	18
	4	SANJOY	KOLKATA	18
	5	VISHAL	TELENGANA	20
	6	PRIYA	ASSAM	19
	7	PURU	BIHAR	18
	8	RIYA	KARNATAKA	19
	NULL	NULL	NULL	NULL

```
CREATE TABLE `assignment5`.`student` (
```

```
    `S_ID` INT NOT NULL AUTO_INCREMENT,
```

```
    `Name` VARCHAR(20) NOT NULL,
```

```
    `Address` VARCHAR(20) NULL,
```

```
    `Age` INT(2) NULL,
```

```
    PRIMARY KEY (`S_ID`)
```

```
);
```

```
INSERT INTO `assignment5`.`student` (`S_ID`, `Name`, `Address`, `Age`) VALUES ('1', 'LOKI', 'DELHI', '19');
```

```
INSERT INTO `assignment5`.`student` (`S_ID`, `Name`, `Address`, `Age`) VALUES ('2', 'KISHAN', 'KERELA', '20');
```

```
INSERT INTO `assignment5`.`student` (`S_ID`, `Name`, `Address`, `Age`) VALUES ('3', 'RISHI', 'ASSAM', '18');
```

```
INSERT INTO `assignment5`.`student` (`S_ID`, `Name`, `Address`, `Age`) VALUES ('4', 'SANJOY', 'KOLKATA', '18');
```

```
INSERT INTO `assignment5`.`student` (`S_ID`, `Name`, `Address`, `Age`) VALUES ('5', 'VISHAL', 'TELENGANA', '20');
```

```
INSERT INTO `assignment5`.`student` (`S_ID`, `Name`, `Address`, `Age`) VALUES ('6', 'PRIYA', 'ASSAM', '19');
```

```
INSERT INTO `assignment5`.`student` (`S_ID`, `Name`, `Address`, `Age`) VALUES ('7', 'PURU', 'BIHAR', '18');
```

```
INSERT INTO `assignment5`.`student` (`S_ID`, `Name`, `Address`, `Age`) VALUES ('8', 'RIYA', 'KARNATAKA', '19');
```

## 1. ii. course

Result Grid			Filter Rows:
	C_ID	S_ID	
▶	1	1	
	2	2	
	2	3	
	3	4	
	1	5	
	4	9	
	5	10	
	4	11	
✱	NULL	NULL	

```
CREATE TABLE `assignment5`.`course` (
  `C_ID` INT NOT NULL,
  `S_ID` INT NOT NULL,
  PRIMARY KEY (`S_ID`)
);
```

```
INSERT INTO `assignment5`.`course` (`C_ID`, `S_ID`) VALUES ('1', '1');
INSERT INTO `assignment5`.`course` (`C_ID`, `S_ID`) VALUES ('2', '2');
INSERT INTO `assignment5`.`course` (`C_ID`, `S_ID`) VALUES ('2', '3');
INSERT INTO `assignment5`.`course` (`C_ID`, `S_ID`) VALUES ('3', '4');
INSERT INTO `assignment5`.`course` (`C_ID`, `S_ID`) VALUES ('1', '5');
INSERT INTO `assignment5`.`course` (`C_ID`, `S_ID`) VALUES ('4', '9');
INSERT INTO `assignment5`.`course` (`C_ID`, `S_ID`) VALUES ('5', '10');
INSERT INTO `assignment5`.`course` (`C_ID`, `S_ID`) VALUES ('4', '11');
```

## 2. salary\_list



Result Grid			
Filter Rows:			
	ID	NAME	SALARY
	1	LOKI	2000
	2	KISHAN	1500
	3	RISHI	2000
	4	SANJOY	3000
	5	VISHAL	4000
	6	PRIYA	1000
	7	PURU	9000
	8	RIYA	2000
	NULL	NULL	NULL

```
CREATE TABLE `assignment5`.`salary_list` (
  `ID` INT NOT NULL AUTO_INCREMENT,
  `NAME` VARCHAR(20) NOT NULL,
  `SALARY` INT NOT NULL,
  PRIMARY KEY (`ID`)
);
```

```
INSERT INTO `assignment5`.`salary_list` (`ID`, `NAME`, `SALARY`) VALUES ('1', 'LOKI', '2000');
INSERT INTO `assignment5`.`salary_list` (`ID`, `NAME`, `SALARY`) VALUES ('2', 'KISHAN', '1500');
INSERT INTO `assignment5`.`salary_list` (`ID`, `NAME`, `SALARY`) VALUES ('3', 'RISHI', '2000');
INSERT INTO `assignment5`.`salary_list` (`ID`, `NAME`, `SALARY`) VALUES ('4', 'SANJOY', '3000');
INSERT INTO `assignment5`.`salary_list` (`ID`, `NAME`, `SALARY`) VALUES ('5', 'VISHAL', '4000');
INSERT INTO `assignment5`.`salary_list` (`ID`, `NAME`, `SALARY`) VALUES ('6', 'PRIYA', '1000');
INSERT INTO `assignment5`.`salary_list` (`ID`, `NAME`, `SALARY`) VALUES ('7', 'PURU', '9000');
INSERT INTO `assignment5`.`salary_list` (`ID`, `NAME`, `SALARY`) VALUES ('8', 'RIYA', '2000');
```

**3. i. employee**

Result Grid



Filter Rows:

	EmpNo	EmpName	DeptId	GenderId
▶	1886	John Snow	101	1
	1889	Amara Giselle	102	2
	1890	Richie Tung	102	1
✱	NULL	NULL	NULL	NULL

**3. ii. department**

Result Grid	Filter
DeptId	DeptName
101	HR
102	IT
NULL	NULL

**3. iv. gender**

Result Grid	Filter Rows:
GenderId	Gender
1	Male
2	Female
NULL	NULL

```
CREATE TABLE `assignment5`.`employee` (
```

```
    `EmpNo` INT NOT NULL,
```

```
    `EmpName` VARCHAR(20) NOT NULL,
```

```
    `DeptId` INT NOT NULL,
```

```
    `GenderId` INT(1) NOT NULL,
```

```
    PRIMARY KEY (`EmpNo`));
```

```
INSERT INTO `assignment5`.`employee` (`EmpNo`, `EmpName`, `DeptId`, `GenderId`) VALUES ('1886', 'John Snow', '101', '1');
```

```
INSERT INTO `assignment5`.`employee` (`EmpNo`, `EmpName`, `DeptId`, `GenderId`) VALUES ('1889', 'Amara Giselle', '102', '2');
```

```
INSERT INTO `assignment5`.`employee` (`EmpNo`, `EmpName`, `DeptId`, `GenderId`) VALUES ('1890', 'Richie Tung', '102', '1');
```

```
CREATE TABLE `assignment5`.`department` (
```

```
    `DeptId` INT NOT NULL,
```

```
    `DeptName` VARCHAR(5) NOT NULL,
```

```
    PRIMARY KEY (`DeptId`));
```

```
INSERT INTO `assignment5`.`department` (`DeptId`, `DeptName`) VALUES ('101', 'HR');
```

```
INSERT INTO `assignment5`.`department` (`DeptId`, `DeptName`) VALUES ('102', 'IT');
```

```
CREATE TABLE `assignment5`.`gender` (
```

```
    `GenderId` INT NOT NULL,
```

```
    `Gender` VARCHAR(10) NOT NULL,
```

```
    PRIMARY KEY (`GenderId`));
```

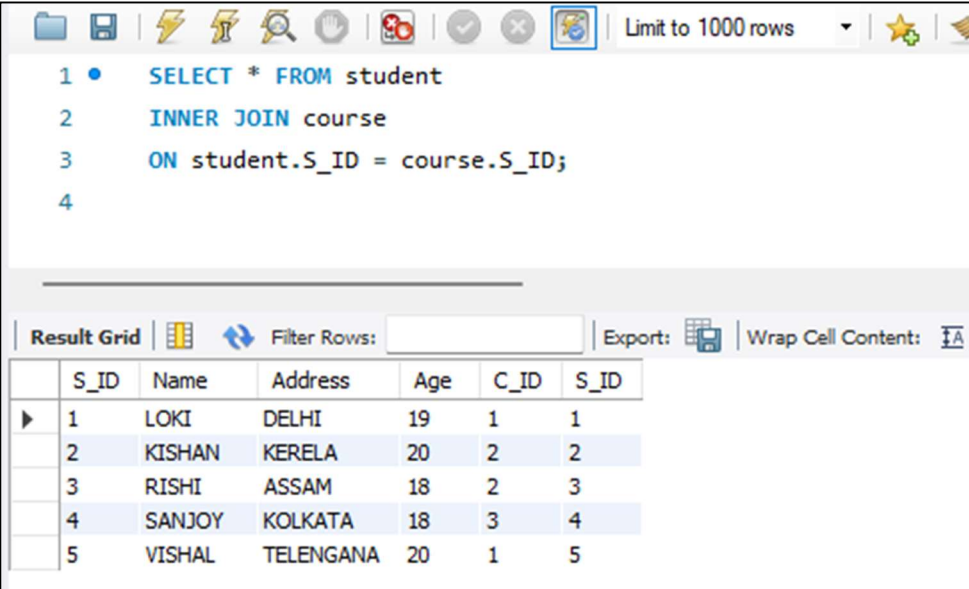
```
INSERT INTO `assignment5`.`gender` (`GenderId`, `Gender`) VALUES ('01', 'Male');
```

```
INSERT INTO `assignment5`.`gender` (`GenderId`, `Gender`) VALUES ('02', 'Female');
```

1. See the given below tables, i.e., Student and Course

i. Write a query to perform the inner join and show the output with all the attributes.

➔ `SELECT * FROM student INNER JOIN course ON student.S_ID = course.S_ID;`



The screenshot shows a database query editor with a toolbar at the top. The query text is as follows:

```

1 • SELECT * FROM student
2   INNER JOIN course
3   ON student.S_ID = course.S_ID;
4

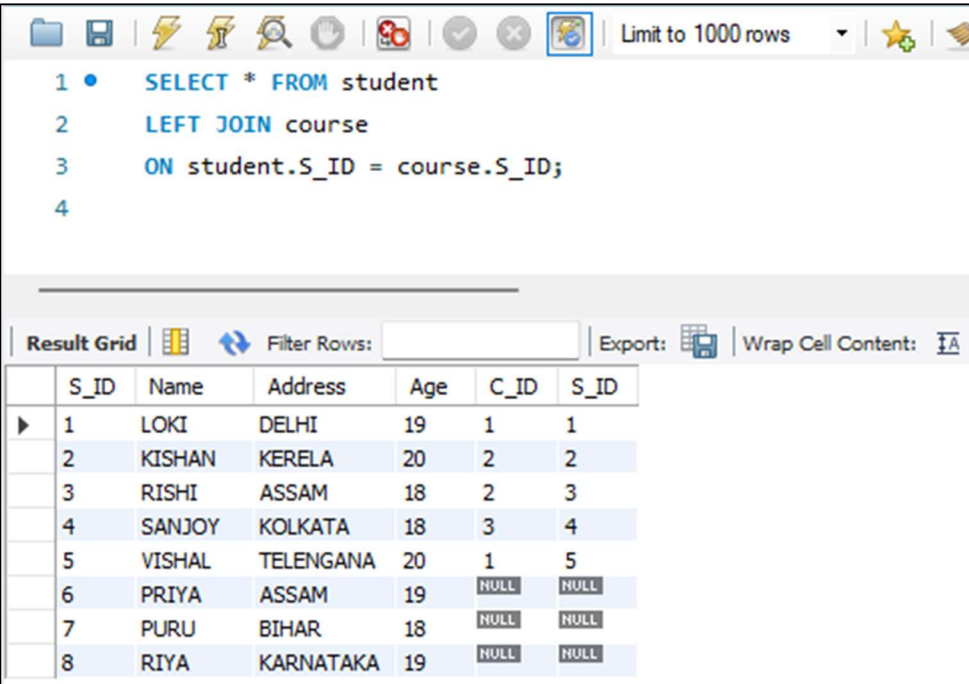
```

Below the query editor is the 'Result Grid' tab, which displays the output of the inner join query. The grid has columns: S\_ID, Name, Address, Age, C\_ID, and S\_ID. The data is as follows:

	S_ID	Name	Address	Age	C_ID	S_ID
▶	1	LOKI	DELHI	19	1	1
	2	KISHAN	KERELA	20	2	2
	3	RISHI	ASSAM	18	2	3
	4	SANJOY	KOLKATA	18	3	4
	5	VISHAL	TELENGANA	20	1	5

ii. Write the query to perform the left join and show the output with all the attributes.

➔ `SELECT * FROM student LEFT JOIN course ON student.S_ID = course.S_ID;`



The screenshot shows a database query editor with a toolbar at the top. The query text is as follows:

```

1 • SELECT * FROM student
2   LEFT JOIN course
3   ON student.S_ID = course.S_ID;
4

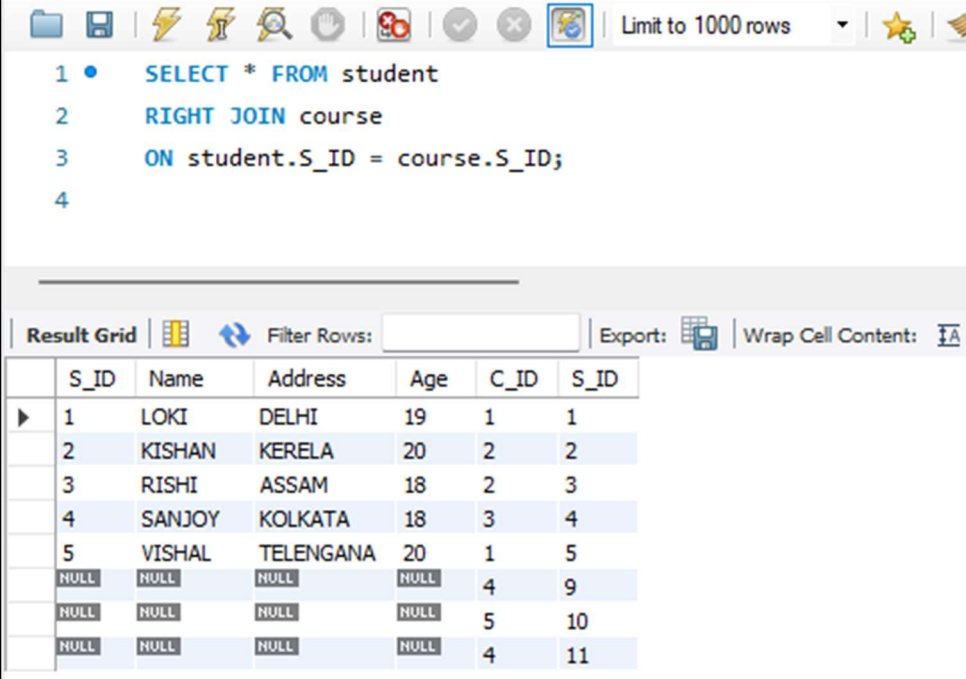
```

Below the query editor is the 'Result Grid' tab, which displays the output of the left join query. The grid has columns: S\_ID, Name, Address, Age, C\_ID, and S\_ID. The data is as follows:

	S_ID	Name	Address	Age	C_ID	S_ID
▶	1	LOKI	DELHI	19	1	1
	2	KISHAN	KERELA	20	2	2
	3	RISHI	ASSAM	18	2	3
	4	SANJOY	KOLKATA	18	3	4
	5	VISHAL	TELENGANA	20	1	5
	6	PRIYA	ASSAM	19	NULL	NULL
	7	PURU	BIHAR	18	NULL	NULL
	8	RIYA	KARNATAKA	19	NULL	NULL

iii. Write the query to perform the right join and show the output with all the attributes.

➔ `SELECT * FROM student RIGHT JOIN course ON student.S_ID = course.S_ID;`



The screenshot shows a database query editor with the following SQL query:

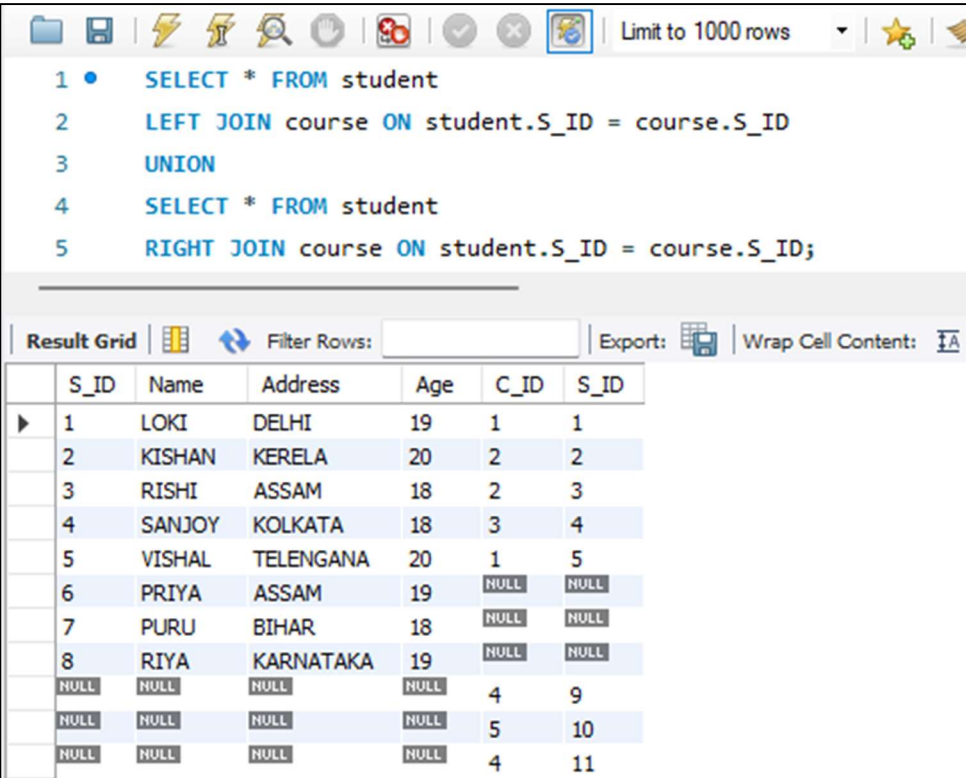
```
1 • SELECT * FROM student
2 RIGHT JOIN course
3 ON student.S_ID = course.S_ID;
4
```

Below the query editor is the 'Result Grid' showing the output of the right join. The grid has columns: S\_ID, Name, Address, Age, C\_ID, and S\_ID. The data is as follows:

	S_ID	Name	Address	Age	C_ID	S_ID
▶	1	LOKI	DELHI	19	1	1
	2	KISHAN	KERELA	20	2	2
	3	RISHI	ASSAM	18	2	3
	4	SANJOY	KOLKATA	18	3	4
	5	VISHAL	TELENGANA	20	1	5
	NULL	NULL	NULL	NULL	4	9
	NULL	NULL	NULL	NULL	5	10
	NULL	NULL	NULL	NULL	4	11

iv. Write the query to perform the full join and show the output with all the attributes.

➔ `SELECT * FROM student LEFT JOIN course ON student.S_ID = course.S_ID UNION  
SELECT * FROM student RIGHT JOIN course ON student.S_ID = course.S_ID;`



The screenshot shows a database query editor with the following SQL query:

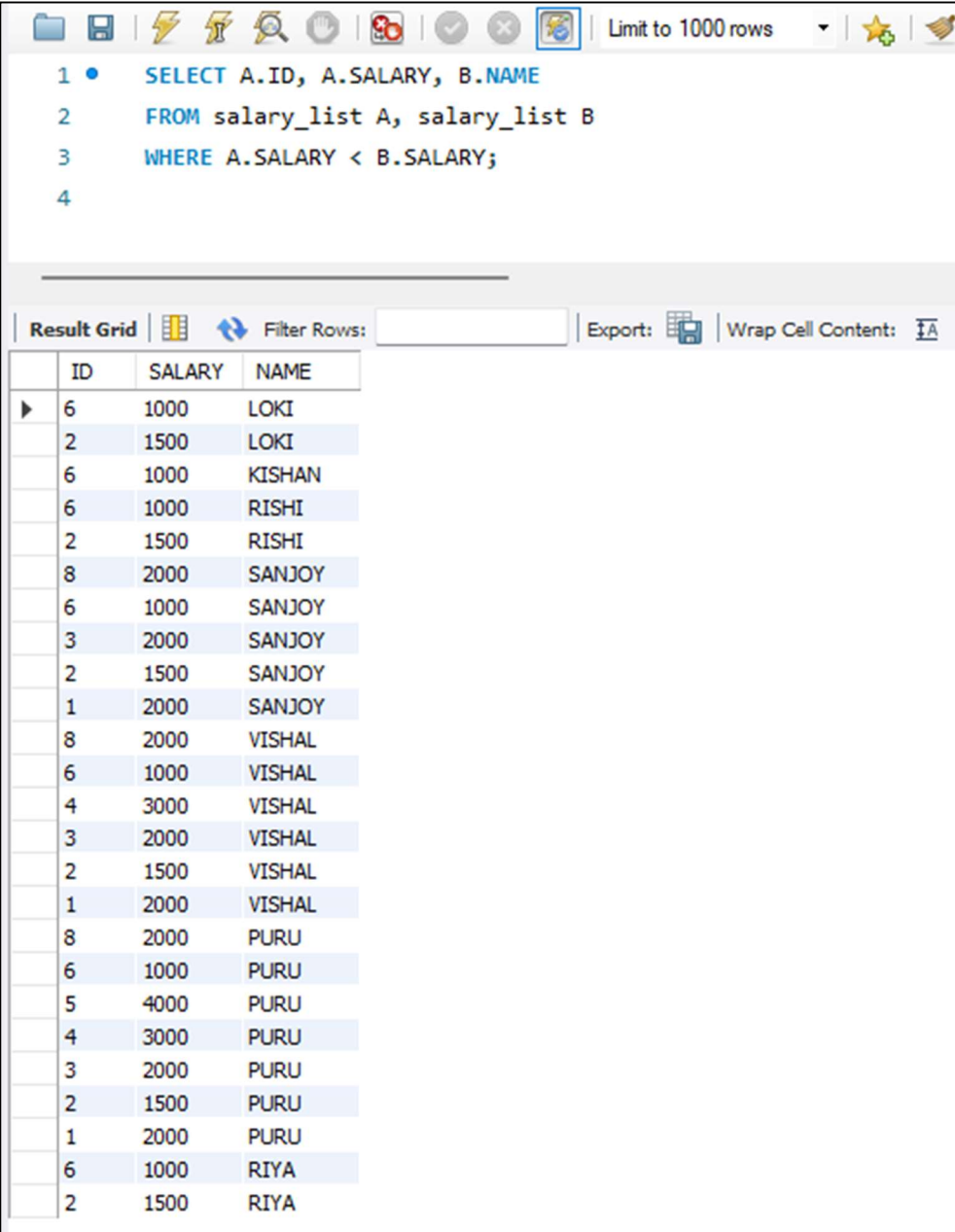
```
1 • SELECT * FROM student
2 LEFT JOIN course ON student.S_ID = course.S_ID
3 UNION
4 SELECT * FROM student
5 RIGHT JOIN course ON student.S_ID = course.S_ID;
```

Below the query editor is the 'Result Grid' showing the output of the full join. The grid has columns: S\_ID, Name, Address, Age, C\_ID, and S\_ID. The data is as follows:

	S_ID	Name	Address	Age	C_ID	S_ID
▶	1	LOKI	DELHI	19	1	1
	2	KISHAN	KERELA	20	2	2
	3	RISHI	ASSAM	18	2	3
	4	SANJOY	KOLKATA	18	3	4
	5	VISHAL	TELENGANA	20	1	5
	6	PRIYA	ASSAM	19	NULL	NULL
	7	PURU	BIHAR	18	NULL	NULL
	8	RIYA	KARNATAKA	19	NULL	NULL
	NULL	NULL	NULL	NULL	4	9
	NULL	NULL	NULL	NULL	5	10
	NULL	NULL	NULL	NULL	4	11

2. See the given below tables i.e., Salary and write the query to perform self-join with the condition "A.SALARY\_LIST < B.SALARY\_LIST", and show the output with ID and SALARY of A and name of B in the output, Where A and B are the alias.

➔ SELECT A.ID, A.SALARY, B.NAME FROM salary\_list A, salary\_list B  
WHERE A.SALARY < B.SALARY;



The screenshot shows a SQL query editor with the following query:

```
1 • SELECT A.ID, A.SALARY, B.NAME
2 FROM salary_list A, salary_list B
3 WHERE A.SALARY < B.SALARY;
4
```

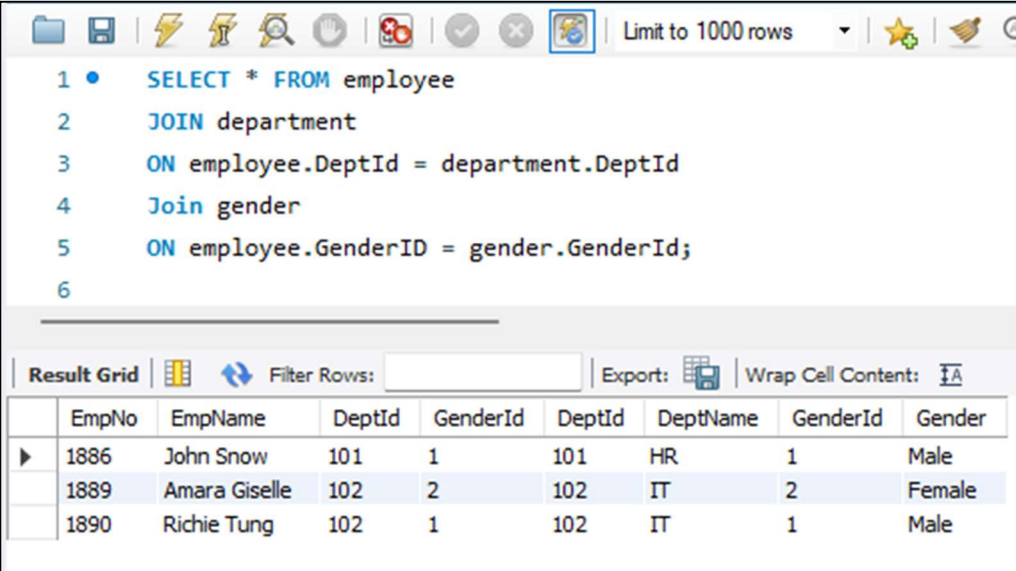
Below the query, the results are displayed in a table with columns ID, SALARY, and NAME. The results are as follows:

ID	SALARY	NAME
6	1000	LOKI
2	1500	LOKI
6	1000	KISHAN
6	1000	RISHI
2	1500	RISHI
8	2000	SANJOY
6	1000	SANJOY
3	2000	SANJOY
2	1500	SANJOY
1	2000	SANJOY
8	2000	VISHAL
6	1000	VISHAL
4	3000	VISHAL
3	2000	VISHAL
2	1500	VISHAL
1	2000	VISHAL
8	2000	PURU
6	1000	PURU
5	4000	PURU
4	3000	PURU
3	2000	PURU
2	1500	PURU
1	2000	PURU
6	1000	RIYA
2	1500	RIYA



3. Write the SQL command to join 3 tables. Tables are given below.

➔ `SELECT * FROM employee`  
`JOIN department ON employee.DeptId = department.DeptId`  
`Join gender ON employee.GenderID = gender.GenderId;`



The screenshot shows a SQL query editor with a toolbar at the top. The query is as follows:

```

1 • SELECT * FROM employee
2 JOIN department
3 ON employee.DeptId = department.DeptId
4 Join gender
5 ON employee.GenderID = gender.GenderId;
6

```

Below the query editor is a 'Result Grid' section. It includes a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The results are displayed in a table with 9 columns: EmpNo, EmpName, DeptId, GenderId, DeptId, DeptName, GenderId, and Gender. There are 3 rows of data.

	EmpNo	EmpName	DeptId	GenderId	DeptId	DeptName	GenderId	Gender
▶	1886	John Snow	101	1	101	HR	1	Male
	1889	Amara Giselle	102	2	102	IT	2	Female
	1890	Richie Tung	102	1	102	IT	1	Male