SQL QUERIES

Team ID – T302

1. Retrieve the names of the top 'n' companies along with their offered salaries,sorted in descending order of salary.

```
SELECT C_Name, Salary
FROM Company
JOIN Job_opening ON Company.C_Id = Job_opening.C_Id
ORDER BY Salary DESC
LIMIT n;

-- Replace 'n' with the desired number of top job salary you want to retrieve
```

2. Retrieve all users who have not applied for any jobs.

```
SELECT Users.U_Id, Users.F_Name, Users.L_Name
FROM Users
LEFT JOIN Application ON Users.U_Id = Application.U_Id
WHERE Application.U_Id IS NULL;
```

3. Retrieve the list of students along with their institutes and grades where the grade is 90% or greater than the total grade.

```
SELECT U.U_Id, U.F_Name,
U.L_Name,E.Grade,E.Total_Grade,E.Institute_Name
FROM Users U
JOIN Education E ON U.U_Id = E.U_Id
WHERE (E.Grade / E.Total_Grade) * 100 >= 90;
```

4. Count the total number of job openings in each location.

```
SELECT Location, COUNT(*) AS Total_Job_Openings
FROM Job_opening
GROUP BY Location;
```

5. Retrieves students who have a number of experiences greater than or equal to the specified limit.

```
SELECT U.U_Id, U.F_Name, U.M_Name, U.L_Name, COUNT(E.U_Id) AS
Num_Experiences
FROM Users U
LEFT JOIN Experience E ON U.U_Id = E.U_Id
GROUP BY U.U_Id, U.F_Name, U.M_Name, U.L_Name
HAVING COUNT(E.U_Id) >= n
```

-- Replace 'n' with the desired number of experiences.

6. List all the data of the currently available job openings.

```
SELECT *
FROM Job_opening
WHERE Appli_deadline > CURRENT_DATE;
```

or

```
SELECT *
FROM Job_opening
WHERE Status = 'OPEN';
```

7. List all companies along with their job openings and the number of applications received for each job opening.

```
SELECT Company.C_Id, Company.C_Name, Job_opening.Job_Roll,
COUNT(Application.U_Id) AS Num_Applications
FROM Company
LEFT JOIN Job_opening ON Company.C_Id = Job_opening.C_Id
LEFT JOIN Application ON Job_opening.C_Id = Application.C_Id
GROUP BY Company.C_Id, Job_opening.Job_Roll;
```

8. Find all users who have a specific skill.

```
SELECT Users.F_Name, Users.L_Name
FROM Users
JOIN Skills ON Users.U_Id = Skills.U_Id
WHERE Skills.Skill_Name = 'Java';
```

9. Retrieves the top students based on their job packages from the 'student' table for accepted job applications with a minimum job salary value specified by the user.

```
SELECT U.U_Id, U.F_Name, U.L_Name, J.Job_Roll, J.Salary
FROM Users U
JOIN Application A ON U.U_Id = A.U_Id
JOIN Job_opening J ON A.C_Id = J.C_Id
WHERE A.Status = 'ACCEPTED' AND J.Salary >= minimum_salary
ORDER BY J.Salary DESC;
```

10. Find the user IDs, first names, last names, and email addresses of users who have applied for a job.

```
SELECT U.U_Id, U.F_Name, U.L_Name, U.U_Email
FROM Users U
INTERSECT
SELECT U.U_Id, U.F_Name, U.L_Name, U.U_Email
FROM Users U
JOIN Application A ON U.U_Id = A.U_Id;
```

11. Find the number of job experiences for each user.

```
SELECT U.U_Id, U.F_Name, U.L_Name, COUNT(E.U_Id) AS
Num_Experiences
FROM Users U
LEFT JOIN Experience E ON U.U_Id = E.U_Id
GROUP BY U.U_Id, U.F_Name, U.L_Name;
```

12. Find users with experience in a particular city and company along with their contact details.

```
SELECT u.U_Email, u.U_Phone, e.Company, e.Location
FROM Users u
INNER JOIN Experience e ON u.U_Id = e.U_Id
WHERE e.Company = 'YourCompanyHere' AND e.Location 'YourCityHere';
```

13. List of applicants to specified company and have minimum of 3 years of experience.

```
SELECT DISTINCT U.U_Id, U.F_Name, U.M_Name, U.L_Name
FROM Users U
JOIN Experience E ON U.U_Id = E.U_Id
JOIN Application A ON U.U_Id = A.U_Id
JOIN Company C ON A.C_Id = C.C_Id
WHERE C.C_Name = 'specified_company_name'
GROUP BY U.U_Id, U.F_Name, U.M_Name, U.L_Name
HAVING DATEDIFF(CURDATE(), MIN(E.Start_date)) >= 1095;

-- Replace 'specified_company_name' with the name of the specified company.
-- Minimum of 3 years (365 * 3) = 1095.
```

14. Retrieve the highest salary offer for each role.

```
SELECT Job_Roll,MAX(Salary) AS Highest_Salary
FROM Job_opening
GROUP BY Job_Roll;
```

15. Retrieve the company names along with the first and last names of users who have been accepted into a job position at those companies.

```
SELECT C.C_Name AS Company_Name, U.F_Name AS First_Name,
U.L_Name AS Last_Name
FROM Users U
INNER JOIN Application A ON U.U_Id = A.U_Id
INNER JOIN Company C ON A.C_Id = C.C_Id
WHERE A.Status = 'ACCEPTED';
```