**Aim**: Sql Join .

**Objectives:**

1. To understand how to retrieve related data from multiple tables using different types of joins in MySQL.
2. To practice writing SQL join queries for common scenarios and conditions.
3. To implement joins effectively to ensure accurate data retrieval from relational tables.

**Tools Used:**

* MySQL Workbench

**Concept:**

* **Joins in SQL:** Joins allow us to retrieve data from multiple tables by connecting records with related values. By using joins, we can efficiently manage relationships in a relational database, accessing data across tables to build complex results.

**Example:**

Using joins to retrieve data:

SELECT student.student\_id, student.student\_name, marks.score

FROM student

JOIN marks ON student.student\_id = marks.student\_id;

**Types of Joins in SQL:**

* **INNER JOIN:** Returns records where there is a match in both tables. Inner joins are used when we need data that exists in both tables.

**Example:** Retrieving records where both student and marks tables have matching student\_ids:

SELECT s.student\_id, s.student\_name, m.score

FROM student s

INNER JOIN marks m ON s.student\_id = m.student\_id;

* **LEFT JOIN (or LEFT OUTER JOIN):** Returns all records from the left table and matched records from the right table. If no match is found, NULL values are returned for columns from the right table.

**Example:** Listing all students and their scores (even if some students don't have scores):

SELECT s.student\_id, s.student\_name, m.score

FROM student s

LEFT JOIN marks m ON s.student\_id = m.student\_id;

* **RIGHT JOIN (or RIGHT OUTER JOIN):** Returns all records from the right table and matched records from the left table. If no match is found, NULL values are returned for columns from the left table.

**Example:** Listing all scores and the associated students (even if some scores don’t have corresponding students):

SELECT s.student\_id, s.student\_name, m.score

FROM student s

RIGHT JOIN marks m ON s.student\_id = m.student\_id;

* **FULL OUTER JOIN:** Returns all records when there is a match in either the left or right table. If there is no match, NULL values are returned for columns where no match is found.

**Example:** *(Note: MySQL doesn’t directly support FULL OUTER JOIN, but we can simulate it using a UNION)*

SELECT s.student\_id, s.student\_name, m.score

FROM student s

LEFT JOIN marks m ON s.student\_id = m.student\_id

UNION

SELECT s.student\_id, s.student\_name, m.score

FROM student s

RIGHT JOIN marks m ON s.student\_id = m.student\_id;

* **CROSS JOIN:** Produces the Cartesian product of the two tables, returning all combinations of records. Useful in scenarios where all pairings of records are needed.

**Example:** Showing all possible student-score combinations:

SELECT s.student\_id, s.student\_name, m.score

FROM student s

CROSS JOIN marks m;

**Problem Statement**

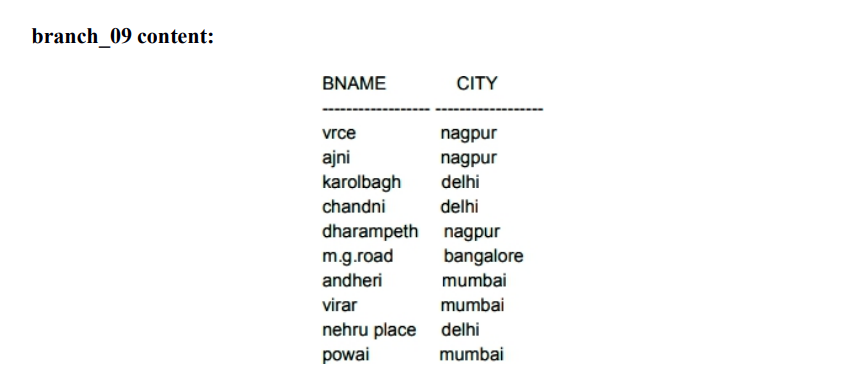
Create 4 tables

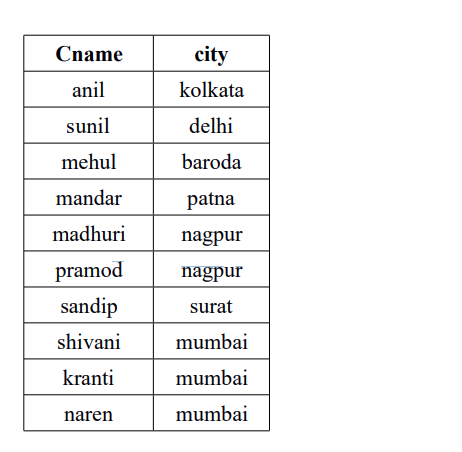
1) CREATE table deposit\_01 (actno varchar2(5), cname varchar2(18), bname varchar2(18), amount number(8,2), adate date); actno is primary key bname and cname are foreign keys

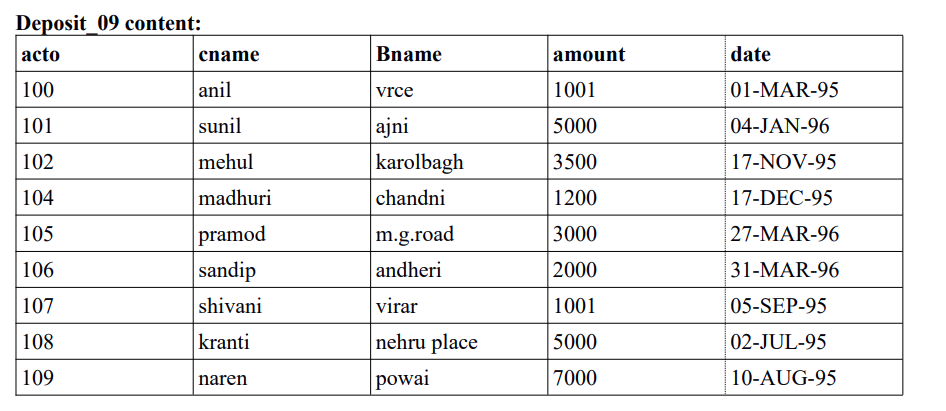
2) CREATE table branch\_01 ( bname varchar2(18), city varchar2(18) ); bname is a primary key

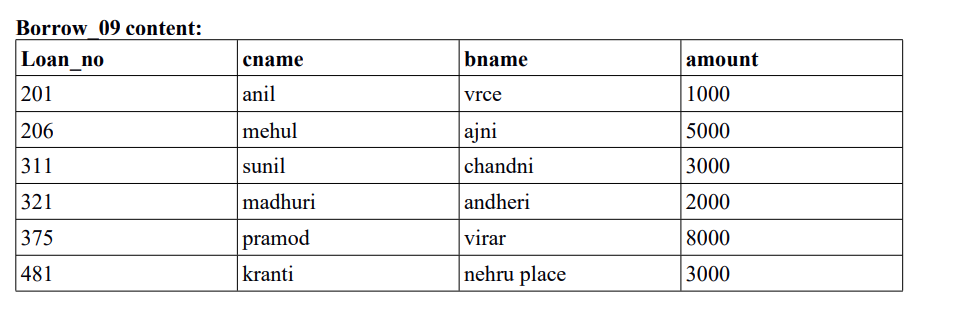
3) CREATE table customer\_01 (cname varchar2(18), city varchar2(18)); cname is primary key

4) CREATE table borrow\_01 ( loan\_no varchar2(5), cname varchar2(18), bname varchar2(18), amount number(8,2) ); loan\_no is primary key bname and cname are foreign keys



**customer\_09 content**





**Basic Queries:**

1) List all data from deposit table.

2) List all data from borrow table.

3) List names of customers living in Nagpur City.

4) List names of borrowers having loan number 206.

5) List names of depositors having amount greater than 4000.

6) List names of customers who opened account after date 1/12/95.

7) List name of the city where the Karol Bagh branch is located.

8) List total loan.

9) List total number of customer cities.

10) Count total number of customers.

11) List maximum loan from VRCE branch.

12) Add 10% interest to all depositors.

13) Add 10% interest to all depositors having VRCE branch.

14) Delete depositors if the branch is Virar and the depositor name is Shivani.

15) Delete customers from Mumbai City.

16) Delete depositor having deposit less than 5000.

**Questions on Sub queries and Joins**

* 1. List names of depositors having the same branch as the branch of **SUNIL**.
  2. List LoanNo and LoanAmount of borrowers having the same branch as the depositor **SUNIL**.
  3. List all depositors living in **NAGPUR**.
  4. List all depositors having deposits in all the branches where **SUNIL** is having an account.
  5. List names of customers having maximum deposit.
  6. List names of customers having maximum deposit among customers living in **Nagpur**.
  7. List the names of branches having the highest number of depositors.
  8. List the highest deposit of the city where the branch of **SUNIL** is located.
  9. List the names of customers having more deposit than the average deposit in their respective branches.
  10. List the names of branches where the number of depositors is less than 2.
  11. Count the number of customers living in the city where the branch is located.
  12. Change the living city of the **VRCE** branch borrowers to **Nagpur**.
  13. Update the deposit of **Anil**. Give him the maximum deposit from depositors living in the city **Nagpur**.
  14. Transfer Rs. 100 from **Anil's** account to **Sunil's** account if both are having the same branch.
  15. Add Rs. 100 to the account of all those depositors who have the highest deposit amount in their respective branches.
  16. Delete branches having deposits from **Nagpur**.
  17. Delete the deposit of **Anil** and **Sunil** if both are living in the same city.
  18. Delete borrowers of branches having the minimum number of customers.
  19. List names of customers who are both depositors and borrowers.
  20. List all customers who are depositors but not borrowers.
  21. List the depositors having the same living city as **Sunil** and the same branch city as **Anil**.
  22. List the depositors having an amount less than 5000 and living in the city as **Shivani**.
  23. List the customers who are borrowers or depositors and have a living city of **Mumbai** and the branch city the same as that of **Sandip**.
  24. List the branch name and branch-wise deposit.
  25. Add 100 to the amount of all depositors having a deposit higher than the average deposit of their branch.
  26. List names of depositors who have the third-highest amount.
  27. List details of depositors according to the ascending order of customer names.