**Day3 Joins**

**What are Joins?**

Joins combine rows from two or more tables based on a related column (in our case: BusID).

**1. INNER JOIN**

*Returns only rows with matching values in both tables.*

*SELECT h.PatientName, h.Disease, s.DriverName, s.Capacity*

*FROM HospitalRecords h*

*INNER JOIN Schoolbus s*

*ON h.BusID = s.BusID;*

Output: Patients with valid bus assignments.

**2. LEFT JOIN (LEFT OUTER JOIN)**

Returns all rows from **HospitalRecords** and matches from **Schoolbus**.

*SELECT h.PatientName, h.Disease, s.DriverName, s.Capacity*

*FROM HospitalRecords h*

*LEFT JOIN Schoolbus s*

*ON h.BusID = s.BusID;*

Output: All patients, bus info if available (NULL if not).

**3. RIGHT JOIN (RIGHT OUTER JOIN)**

Returns all rows from **Schoolbus** and matches from **HospitalRecords**.

*SELECT h.PatientName, h.Disease, s.DriverName, s.Capacity*

*FROM HospitalRecords h*

*RIGHT JOIN Schoolbus s*

*ON h.BusID = s.BusID;*

Output: All buses, patient info if available (NULL if not).

**4. FULL OUTER JOIN**

*Returns all rows from both tables.*

*SELECT h.PatientName, h.Disease, s.DriverName, s.Capacity*

*FROM HospitalRecords h*

*FULL OUTER JOIN Schoolbus s*

*ON h.BusID = s.BusID;*

Output: All patients + all buses, matched where possible.

**5. CROSS JOIN**

Cartesian product — every patient with every bus.

SELECT h.PatientName, s.DriverName

FROM HospitalRecords h

CROSS JOIN Schoolbus s;

**Output: All combinations of patients and buses.**

**Key Use-Cases**

* **INNER JOIN** → Only patients with assigned buses.
* **LEFT JOIN** → All patients, even without buses.
* **RIGHT JOIN** → All buses, even without patients.
* **FULL JOIN** → Everything from both tables.
* **CROSS JOIN** → Testing or special scenarios.

**🛠 SQL Code: Create BusAssignment Table**

CREATE TABLE BusAssignment (

AssignmentID INT PRIMARY KEY,

PatientID INT,

BusID INT,

PickupPoint VARCHAR(100),

DropPoint VARCHAR(100),

FOREIGN KEY (PatientID) REFERENCES HospitalRecords(PatientID),

FOREIGN KEY (BusID) REFERENCES Schoolbus(BusID)

);

**Insert Sample Data**

INSERT INTO BusAssignment (AssignmentID, PatientID, BusID, PickupPoint, DropPoint) VALUES

(1, 101, 1, 'Sector 21', 'City Hospital'),

(2, 102, 2, 'MG Road', 'City Hospital'),

(3, 103, 3, 'Raj Nagar', 'City Hospital'),

(4, 104, 1, 'DLF Phase 2', 'City Hospital'),

(5, 105, 2, 'Saket', 'City Hospital'),

(6, 106, NULL, 'Connaught Place', 'City Hospital'), -- Patient not yet assigned a bus

(7, 107, 4, 'Noida Sec 15', 'City Hospital');

**🔗 Why this helps in JOINs?**

Now you can run JOIN queries such as:

**INNER JOIN Example**

SELECT h.PatientName, s.DriverName, b.PickupPoint, b.DropPoint

FROM BusAssignment b

INNER JOIN HospitalRecords h ON b.PatientID = h.PatientID

INNER JOIN Schoolbus s ON b.BusID = s.BusID;

👉 Shows patients, their assigned bus driver, and pickup/drop points.

**LEFT JOIN Example**

SELECT h.PatientName, s.DriverName, b.PickupPoint

FROM HospitalRecords h

LEFT JOIN BusAssignment b ON h.PatientID = b.PatientID

LEFT JOIN Schoolbus s ON b.BusID = s.BusID;

👉 Shows **all patients**, even those not yet assigned a bus (NULL for bus details).

Perfect 👍 then let’s refine **Day3 Joins Guide** using a **proper relational model** with three tables:

* **Schoolbus** → (BusID, DriverName, Capacity)
* **HospitalRecords** → (PatientID, PatientName, Age, Gender, Disease, AdmissionDate, DoctorName)
* **BusAssignment** → (AssignmentID, PatientID, BusID, PickupPoint, DropPoint)

This way, BusAssignment acts as a **bridge** table linking patients to buses.

**📘 Day3 Joins Guide (with BusAssignment)**

**What are Joins?**

Joins combine rows from two or more tables based on a related column.  
Here, **BusAssignment** links patients (PatientID) with buses (BusID).

**1. INNER JOIN**

Patients with assigned buses and driver details.

*SELECT h.PatientName, h.Disease, s.DriverName, s.Capacity, b.PickupPoint*

*FROM BusAssignment b*

*INNER JOIN HospitalRecords h ON b.PatientID = h.PatientID*

*INNER JOIN Schoolbus s ON b.BusID = s.BusID;*

Output: Only patients with a valid bus assignment.

**2. LEFT JOIN**

All patients, even those without a bus assignment.

SELECT h.PatientName, h.Disease, s.DriverName, b.PickupPoint

FROM HospitalRecords h

LEFT JOIN BusAssignment b ON h.PatientID = b.PatientID

LEFT JOIN Schoolbus s ON b.BusID = s.BusID;

Output: Every patient → if no bus assigned, bus details show NULL.

**3. RIGHT JOIN**

All buses, even those without any patients.

*SELECT s.BusID, s.DriverName, h.PatientName, b.PickupPoint*

*FROM Schoolbus s*

*RIGHT JOIN BusAssignment b ON s.BusID = b.BusID*

*RIGHT JOIN HospitalRecords h ON b.PatientID = h.PatientID;*

Output: Every bus → shows assigned patients if available, else NULL.

**4. FULL OUTER JOIN**

All patients and all buses, matched where possible.

SELECT h.PatientName, s.DriverName, b.PickupPoint

FROM BusAssignment b

FULL OUTER JOIN HospitalRecords h ON b.PatientID = h.PatientID

FULL OUTER JOIN Schoolbus s ON b.BusID = s.BusID;

Output: Complete picture of patients, buses, and assignments.

**5. CROSS JOIN**

Every patient paired with every bus (for testing).

SELECT h.PatientName, s.DriverName

FROM HospitalRecords h

CROSS JOIN Schoolbus s;

Output: Cartesian product → each patient with every bus.

**Key Use Cases**

* **INNER JOIN** → See only patients with bus details.
* **LEFT JOIN** → Ensure no patient is missed, even without a bus.
* **RIGHT JOIN** → Ensure no bus is missed, even if empty.
* **FULL OUTER JOIN** → Show everything from both.
* **CROSS JOIN** → Special cases/testing.