

SQL Join

SQL JOIN - বিস্তারিত ব্যাখ্যা (বাংলায়)

টেবিল ডেটা:

Students Table:

student_id student_name class

1	Alice	4DM
2	Bob	4DM
3	Charlie	4DM
4	David	4DM

Courses Table:

course_id course_name student_id

101	Database	1
102	Math	1
103	Physics	2
104	Chemistry	NULL

1. INNER JOIN

কোড:

```
SELECT s.student_name, c.course_name  
FROM students s  
INNER JOIN courses c ON s.student_id = c.student_id;
```

ব্যাখ্যা:

- INNER JOIN শব্দমাত্র সেই rows return করে যেগুলো উভয় table-এ match করে
- যে student-দের course আছে শুধু তাদের দেখাবে

- যে student-দের course নেই (যেমন Charlie, David) তারা আসবে না
- যে course-এর student নেই (Chemistry) সেটাও আসবে না

আউটপুট:

student_name course_name

Alice Database

Alice Math

Bob Physics

মনে রাখার কথা: শুধু matching data দেখায়

2. LEFT JOIN (LEFT OUTER JOIN)

কোড:

```
SELECT s.student_name, c.course_name
FROM students s
LEFT JOIN courses c ON s.student_id = c.student_id;
```

ব্যাখ্যা:

- **LEFT JOIN** বাম পাশের table-এর (students) **সব rows** return করে
- ডান পাশের table-এ (courses) match না থাকলে **NULL** দেখায়
- **সব students** দেখাবে, course থাক বা না থাক
- যে student-দের course নেই তাদের course_name column-এ NULL আসবে

আউটপুট:

student_name course_name

Alice Database

Alice Math

student_name course_name

Bob Physics

Charlie NULL

David NULL

মনে রাখার কথা: বাম table সম্পূর্ণ + ডান table থেকে matching

3. RIGHT JOIN (RIGHT OUTER JOIN)

কোড:

```
SELECT s.student_name, c.course_name  
FROM students s  
RIGHT JOIN courses c ON s.student_id = c.student_id;
```

ব্যাখ্যা:

- **RIGHT JOIN** ডান পাশের table-এর (courses) **সব rows** return করে
- বাম পাশের table-এ (students) match না থাকলে **NULL** দেখায়
- **সব courses** দেখাবে, student assign থাক বা না থাক
- যে course-এ student নেই (Chemistry) সেটার student_name NULL আসবে

আউটপুট:

student_name course_name

Alice Database

Alice Math

Bob Physics

NULL Chemistry

মনে রাখার কথা: ডান table সম্পূর্ণ + বাম table থেকে matching

4. FULL OUTER JOIN (MySQL-এ UNION দিয়ে)

কোড:

```
SELECT s.student_name, c.course_name  
FROM students s  
LEFT JOIN courses c ON s.student_id = c.student_id  
UNION  
SELECT s.student_name, c.course_name  
FROM students s  
RIGHT JOIN courses c ON s.student_id = c.student_id;
```

ব্যাখ্যা:

- **FULL OUTER JOIN** উভয় table-এর সব rows return করে
- MySQL-এ direct FULL OUTER JOIN নেই, তাই LEFT + RIGHT JOIN-কে UNION করতে হয়
- যেখানে match নেই সেখানে NULL দেখায়
- সব students এবং সব courses একসাথে দেখাবে

আউটপুট:

student_name course_name

Alice Database

Alice Math

Bob Physics

Charlie NULL

David NULL

NULL Chemistry

মনে রাখার কথা: উভয় table থেকে সব data

5. CROSS JOIN

কোড:

```
SELECT s.student_name, c.course_name  
FROM students s  
CROSS JOIN courses c;
```

ব্যাখ্যা:

- **CROSS JOIN** দুটি table-এর **Cartesian Product** তৈরি করে
- প্রথম table-এর প্রতিটি row, দ্বিতীয় table-এর প্রতিটি row-এর সাথে combine হয়
- কোনো condition লাগে না
- Students = 4 rows, Courses = 4 rows → Result = $4 \times 4 = 16$ rows

আউটপুট:

student_name course_name

Alice Database

Alice Math

Alice Physics

Alice Chemistry

Bob Database

Bob Math

Bob Physics

Bob Chemistry

Charlie Database

Charlie Math

Charlie Physics

student_name course_name

Charlie Chemistry

David Database

David Math

David Physics

David Chemistry

মনে রাখার কথা: সব possible combinations

6. SELF JOIN

কোড:

```
SELECT s1.student_name AS student1, s2.student_name AS student2  
FROM students s1  
INNER JOIN students s2 ON s1.student_id < s2.student_id;
```

ব্যাখ্যা:

- **SELF JOIN** একই table-কে নিজের সাথে join করে
- একই table-কে দুইবার use করতে হয় (s1, s2 alias দিয়ে)
- এখানে সব student-দের pair বানানো হচ্ছে
- s1.student_id < s2.student_id মানে duplicate pair avoid করা (যেমন Alice-Bob থাকলে Bob-Alice লাগবে না)

আউটপুট:

student1 student2

Alice Bob

Alice Charlie

Alice David

student1 student2

Bob Charlie

Bob David

Charlie David

মনে রাখার কথা: একই table থেকে দুই আলাদা row compare করা

7. NATURAL JOIN

কোড:

```
SELECT *  
FROM students  
NATURAL JOIN courses;
```

ব্যাখ্যা:

- **NATURAL JOIN** automatically সেই column-গুলোতে join করে যেগুলোর নাম একই
- এখানে student_id column উভয় table-এ আছে, তাই এটাতে join হবে
- ON clause লেখার দরকার নেই
- **⚠️ সতর্কতা:** একই নামের অনাকাঙ্ক্ষিত column থাকলে ভুল result আসতে পাবে

আউটপুট:

student_id student_name class course_id course_name

1	Alice	4DM 101	Database
1	Alice	4DM 102	Math
2	Bob	4DM 103	Physics

মনে রাখার কথা: Automatic join, same name column-এ

8. JOIN USING

কোড:

```
SELECT *  
FROM students s  
INNER JOIN courses c USING(student_id);
```

ব্যাখ্যা:

- **USING** clause specific column name দিয়ে join করে
- ON s.student_id = c.student_id এর সংক্ষিপ্ত রূপ
- শুধুমাত্র তখনই ব্যবহার করা যায় যখন উভয় table-এ column-এর নাম একই
- আউটপুটে student_id একবারই আসবে (দুইবার না)

আউটপুট:

student_id	student_name	class	course_id	course_name
1	Alice	4DM	101	Database
1	Alice	4DM	102	Math
2	Bob	4DM	103	Physics

মনে রাখার কথা: USING = ON-এর সংক্ষিপ্ত form

9. ANTI JOIN (যাদের match নেই)

কোড:

```
SELECT s.student_name  
FROM students s  
LEFT JOIN courses c ON s.student_id = c.student_id  
WHERE c.course_id IS NULL;
```

ব্যাখ্যা:

- ANTI JOIN সেই rows খুঁজে বের করে যেগুলোর match নেই

- LEFT JOIN করে তারপর WHERE column IS NULL দিয়ে filter করা
- এখানে যে students-দের কোনো course নেই তাদের খুঁজছি
- Charlie এবং David-এর কোনো course নেই

আউটপুট:

student_name

Charlie

David

মনে রাখার কথা: যাদের বিভিন্ন table-এ কোনো entry নেই

10. SEMI JOIN (যাদের match আছে)

কোড:

```
SELECT s.student_name
FROM students s
WHERE s.student_id IN (SELECT student_id FROM courses WHERE student_id IS NOT NULL);
```

ব্যাখ্যা:

- SEMI JOIN সেই rows return করে যেগুলোর match আছে
- Subquery দিয়ে courses table-এ যে student_id আছে সেগুলো check করা
- শুধু students table থেকে data আসবে (courses থেকে না)
- যে students-দের অন্তত একটা course আছে তাদের দেখাবে

আউটপুট:

student_name

Alice

Bob

মনে রাখার কথা: যাদের বিভিন্ন table-এ entry আছে (শুধু প্রথম table-এর data)

JOIN তুলনা চাট:

JOIN Type	বাম Table	ডান Table	Match না থাকলে
INNER	শুধু matching	শুধু matching	দেখায় না
LEFT	সব rows	শুধু matching NULL	দেখায় (ডানে)
RIGHT	শুধু matching	সব rows	NULL দেখায় (বামে)
FULL OUTER	সব rows	সব rows	উভয় দিকে NULL
CROSS	সব rows	সব rows	সব combination

কখন কোন JOIN ব্যবহার করবেন?

- **INNER JOIN:** যখন শুধু matching data চাই
- **LEFT JOIN:** বাম table-এর সব data চাই, ডানের match থাক বা না থাক
- **RIGHT JOIN:** ডান table-এর সব data চাই
- **FULL OUTER JOIN:** উভয় table-এর সব data চাই
- **CROSS JOIN:** সব possible combinations চাই
- **SELF JOIN:** একই table-এর rows compare করতে চাই
- **ANTI JOIN:** যাদের match নেই তাদের খুঁজতে
- **SEMI JOIN:** যাদের match আছে শুধু তাদের (duplicate ছাড়া)