

Lecture Notes – Cross & InnerJoin Queries

Section 1: Lecture Summary

The lecture covers syntax and practical queries for **CROSS JOIN** and **INNER JOIN** in MySQL. It demonstrates pairing all rows between tables using **CROSS JOIN** and using **JOIN** with **ON** conditions beyond equality, including greater-than comparisons. Examples use the eCommerceDB to pair products with payments where product price exceeds payment amount and to find orders where payment amount is less than order total amount.

Section 2: Key Concepts and Explanations

CROSS JOIN produces a Cartesian product, pairing every row from the first table with every row from the second table, resulting in rows equal to the product of the row counts (e.g., 20 employees × 4 departments = 80 rows).

INNER JOIN (or just **JOIN**) filters rows based on an **ON** condition and can use operators like greater than (>), not just equality (=).

The **ON** clause allows flexible conditions, such as comparing values across unrelated tables to find mismatches like higher product prices versus payment amounts or insufficient payments relative to order totals.

JOIN, **INNER JOIN**, and **CROSS JOIN** yield the same results when an **ON** condition is applied, as the condition filters the full Cartesian product.

Section 3: Example Code and Use Cases

Query to pair every **product** with every **payment** where product **price** > payment **amount** (shows products more expensive than certain payments, e.g., 28 rows result):

```
SELECT p.ProductName, p.Price, pay.Amount
FROM Products p
INNER JOIN Payments pay ON p.Price > pay.Amount
ORDER BY p.ProductName;
```

Query to find **orders** where **payment amount** < **order total amount** (identifies underpaid orders):

```
SELECT o.OrderID, o.TotalAmount, pay.Amount  
FROM Orders o  
INNER JOIN Payments pay ON o.OrderID = pay.OrderID AND pay.Amount <  
o.TotalAmount;
```

Section 4: Key Takeaways

- Use **SELECT * FROM table1 CROSS JOIN table2** for all possible row pairings.
- Apply **ON** conditions with **>**, **<**, or other operators in **JOIN/INNER JOIN/CROSS JOIN** for filtered results without equality.
- Practice these on eCommerceDB tables like **Products**, **Payments**, **Orders** to compare prices, amounts, and totals.