

Lecture Notes – Natural Join Queries

Section 1: Lecture Summary

This lecture covers **NATURAL JOIN** queries, a method for combining multiple tables based on common columns. The focus is on practical query writing using NATURAL JOIN to retrieve meaningful results from related tables. Three main examples demonstrate how to structure queries that join employees with projects, orders with payments, and products with reviews. The lecture emphasizes understanding which tables contain required data and how to filter results appropriately.

Section 2: Key Concepts and Explanations

NATURAL JOIN Syntax and Purpose

A NATURAL JOIN combines two tables by automatically matching columns that share the same name. The syntax is straightforward: `SELECT columns FROM table1 NATURAL JOIN table2`. This join type is most effective when one table contains a primary key and the related table contains the corresponding foreign key with identical column names.

When to Use NATURAL JOIN

NATURAL JOIN is applicable when tables have common columns that logically relate data. This eliminates the need to explicitly specify join conditions, as the database automatically identifies matching column names between tables.

Multiple Table Joins

Multiple tables can be joined sequentially in a single query. For example, connecting employees, their project assignments, and project details requires chaining NATURAL JOINS together: `FROM employees NATURAL JOIN employees_projects NATURAL JOIN projects`.

****Filtering with WHERE Clauses****

Results from joined tables can be filtered using WHERE clauses to retrieve specific subsets of data. This is essential when you need only certain records, such as reviews with a specific rating or projects assigned to particular employees.

Section 3: Example Code and Use Cases

****Query 1: Employee Project Assignments****

This query retrieves which employee is assigned to which project along with project details.

```
SELECT EmpID, FirstName, LastName, ProjectID, ProjectName
FROM Employees
NATURAL JOIN EmployeeProjects
NATURAL JOIN Projects;
```

This query joins the Employees table with EmployeeProjects to identify assignments, then joins with Projects to include project names. The result shows all 20 employee-project relationships with complete employee and project information.

****Query 2: Product Reviews with Specific Rating****

This query displays products that have received reviews with a rating of 5.

```
SELECT ProductID, ProductName, Rating, ReviewDate
FROM Products
NATURAL JOIN Reviews
WHERE Rating = 5
ORDER BY ReviewDate DESC;
```

The query filters reviews to show only those with a rating of 5 and sorts results by review date in descending order. The common column (ProductID) automatically links products to their corresponding reviews.

****Query 3: Employee and Department Information****

This query connects employees with their department details.

```
SELECT EmpID, FirstName, LastName, DeptID, DeptName, Location  
FROM Employees  
NATURAL JOIN Departments;
```

The DeptID column serves as the linking point between the Employees and Departments tables, automatically matching records where department IDs are identical.

Section 4: Key Takeaways

NATURAL JOIN is effective for joining tables with common column names, reducing the need for explicit join condition syntax. Understanding which tables contain required data is the first step in constructing effective queries. Multiple tables can be combined in sequence by chaining NATURAL JOINS together. WHERE clauses add filtering capability to retrieve specific subsets from joined results. NATURAL JOIN works best in databases with consistent naming conventions where foreign keys share names with their corresponding primary keys.