Lecture Notes – Set Operations

Section 1: Lecture Summary

Set operations in SQL include **union**, **intersection**, **except** (difference), and **union all**, derived from mathematical set operations on rows from query results. These combine results from multiple queries, with **union** merging sets and removing duplicates, **union all** merging without removing duplicates, **intersection** taking only common rows, and **except** returning rows from the first query not in the second.

Section 2: Key Concepts and Explanations

Set operations treat query results as sets of rows, similar to mathematical sets A and B. For **union** (A ∪ B), it combines all unique elements (e.g., from A={1,2,3,4,5,6} and B={4,5,6,7,8,9}, result={1,2,3,4,5,6,7,8,9}). **Union all** includes duplicates (e.g., result={1,2,3,4,5,6,4,5,6,7,8,9}). **Intersection** (A ∩ B) returns only common elements (e.g., {4,5,6}). **Except** (A - B) subtracts common elements from A (e.g., {1,2,3}). Operations require compatible data types and structures between queries, allowing complex queries to be broken into parts for combination.

Section 3: Example Code and Use Cases

Using companyDB, examples illustrate set operations on compatible row sets like EmpID and DeptID.

```
-- Union: Employees and projects by DeptID (unique rows)
SELECT DeptID FROM Employees
UNION
SELECT DeptID FROM Projects;

-- Union All: Includes duplicates if any
SELECT DeptID FROM Employees
UNION ALL
SELECT DeptID FROM Projects;

-- Intersect: Common DeptID in Employees and Projects
SELECT DeptID FROM Employees
INTERSECT
SELECT DeptID FROM Projects;
```

```
-- Except: DeptID in Employees but not Projects
SELECT DeptID FROM Employees
EXCEPT
SELECT DeptID FROM Projects;
```

These combine results vertically, assuming matching column counts and types (e.g., both select single int column).

Section 4: Key Takeaways

Set operations (**union**, **union all**, **intersect**, **except**) merge query results like mathematical sets, removing duplicates except in **union all**. Compatible data types are required; use to simplify complex queries by combining sub-results.