

Lecture Notes – Simple View

Section 1: Lecture Summary

Views are saved SQL queries that act as stored virtual tables, allowing reuse in other queries without storing data separately. They simplify complex queries, reduce repetition for frequent queries, add a security layer by restricting access to specific columns or data, and remain dynamic with fresh data execution each time.

Section 2: Key Concepts and Explanations

Views function like virtual tables from saved SELECT queries, enabling simpler queries on top of them. Users can query views as tables, selecting columns, using WHERE clauses, but only access defined columns and rows for security. **Simple views** are created from basic SELECT statements on one or more tables, hiding unnecessary columns like salary or hire date. Views simplify complexity by encapsulating detailed logic, support user restrictions to prevent full table access, and execute dynamically for up-to-date results.

Section 3: Example Code and Use Cases

Using **companyDB** schema:

```
CREATE VIEW employees_info AS
SELECT EmpID, FirstName, LastName, JobTitle
FROM Employees;
```

This creates a **simple view** with only four columns from **Employees** table, excluding DeptID, HireDate, Salary.

Query the view:

```
SELECT * FROM employees_info;
```

Retrieves EmpID, FirstName, LastName, JobTitle from the underlying **Employees** table via the view.

Add filter:

```
SELECT * FROM employees_info
WHERE JobTitle = 'Manager';
```

Filters results using the view like a table.

Section 4: Key Takeaways

Views save SQL queries as virtual tables for reuse, simplification, security, and dynamic data access. Simple views use basic SELECT from one table; create with CREATE VIEW, query like tables.