

Lecture Notes – CTAS

Section 1: Lecture Summary

CTAS stands for **Create Table As Select**, a method to create a physical table from the result of a SELECT query, acting as a one-time snapshot of data. Unlike views, which are virtual and update automatically, CTAS produces a static table that does not include primary keys, foreign keys, indexes, or auto-increment features, only data and data types. It is useful for summarized data that changes infrequently, requiring manual recreation by dropping and recreating the table when updates are needed.

Section 2: Key Concepts and Explanations

CTAS executes a SELECT query and stores its result as a new physical table, similar to a materialized view but as a real table rather than a view. The table captures the query's data and preserves data types but excludes constraints like primary keys, foreign keys, and indexes. Syntax involves **CREATE TABLE table_name AS (SELECT statement)**, where the SELECT can include columns, conditions, GROUP BY, or other clauses from existing tables. The resulting table behaves like any other table for subsequent queries but remains static until manually refreshed.

Section 3: Example Code and Use Cases

Using the companyDB schema, create a table of high-salary employees from the **Employees** table:

```
CREATE TABLE HighSalaryEmployees AS
(SELECT *
 FROM Employees
 WHERE Salary > 60000);
```

This creates **HighSalaryEmployees** with all columns from **Employees** but only rows where **Salary** exceeds 60000, including **EmpID**, **FirstName**, **LastName**, **JobTitle**, **DeptID**, **HireDate**, and **Salary**.

Query the new table for summary statistics:

```
SELECT MIN(Salary) AS MinSalary, MAX(Salary) AS MaxSalary  
FROM HighSalaryEmployees;
```

This returns the minimum and maximum salaries among high-salary employees, demonstrating use as a snapshot for analysis.

Section 4: Key Takeaways

CTAS creates physical tables from SELECT results for static snapshots, ideal for infrequently changing summarized data. It copies data and types but omits keys, indexes, and auto-increments. Refresh by dropping and recreating; use like regular tables in queries for simplicity and performance on stable data.