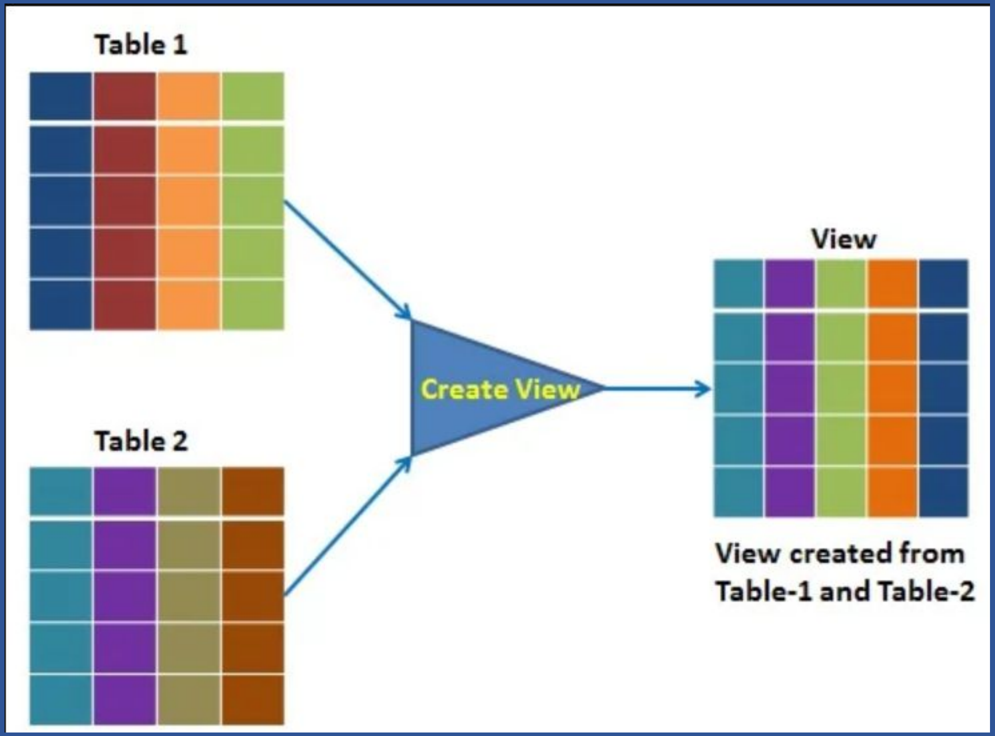
Grow Data Skills

# AGENDA

- WINDOW FUNCTIONS
- COMMON TABLE EXPRESSIONS(CTE)
- VIEWS
- STORED PROCEDURES
- INDEXES

# VIEWS IN SQL

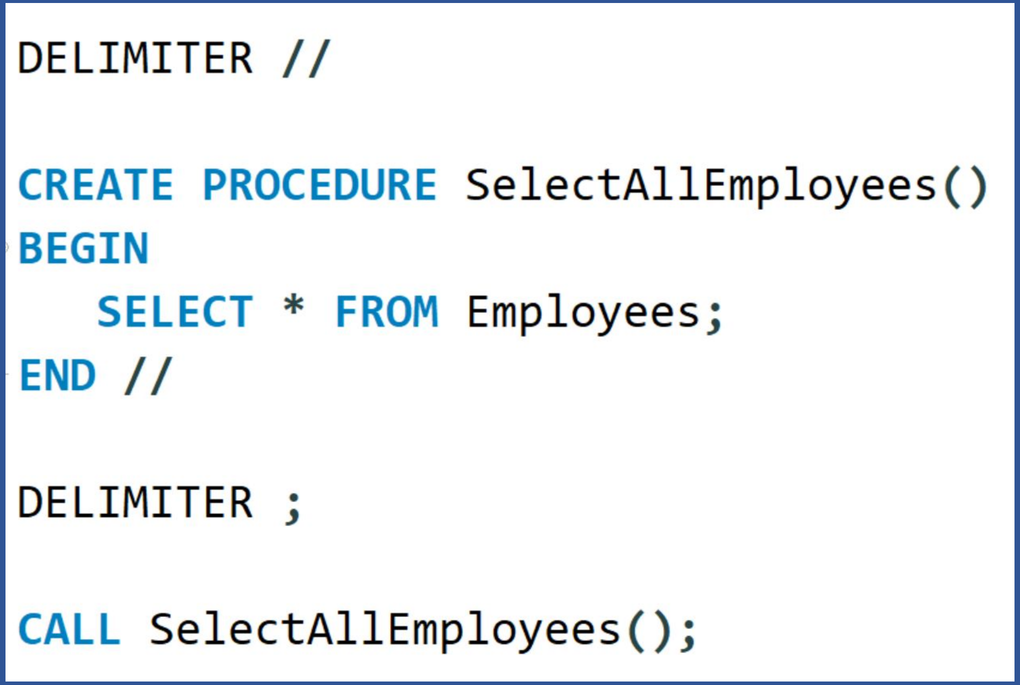
- View is a virtual table based on the result-set of an SQL statement.
- A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.
- You can add SQL statements and functions to a view and present the data as if the data were coming from one single table.
- A view is created with the CREATE VIEW statement.



# STORED PROCEDURES IN SQL

Grow Data Skills

- A stored procedure is a prepared SQL code that you can save, so the code can be reused over and over again.
- So if you have an SQL query that you write over and over again, save it as a stored procedure, and then just call it to execute it.
- You can also pass parameters to a stored procedure, so that the stored procedure can act based on the parameter value(s) that is passed.



```sql
DELIMITER //

CREATE PROCEDURE SelectAllEmployees()
BEGIN
   SELECT * FROM Employees;
END //

DELIMITER ;

CALL SelectAllEmployees();
```

# INDEXES IN SQL

Grow Data Skills

- Indexes are references to data which are used by queries to find data from tables quickly.
- It cannot be viewed by the users and just used to speed up the database access.
- Without index, the query engine checks every row in the table from beginning till the end- **TABLE SCAN**

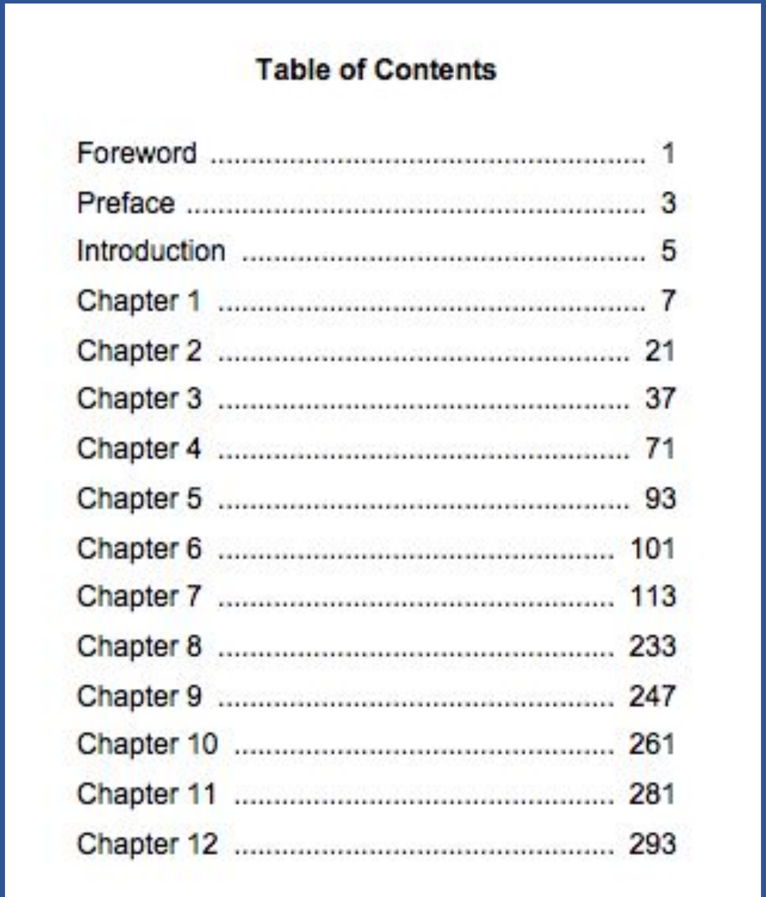
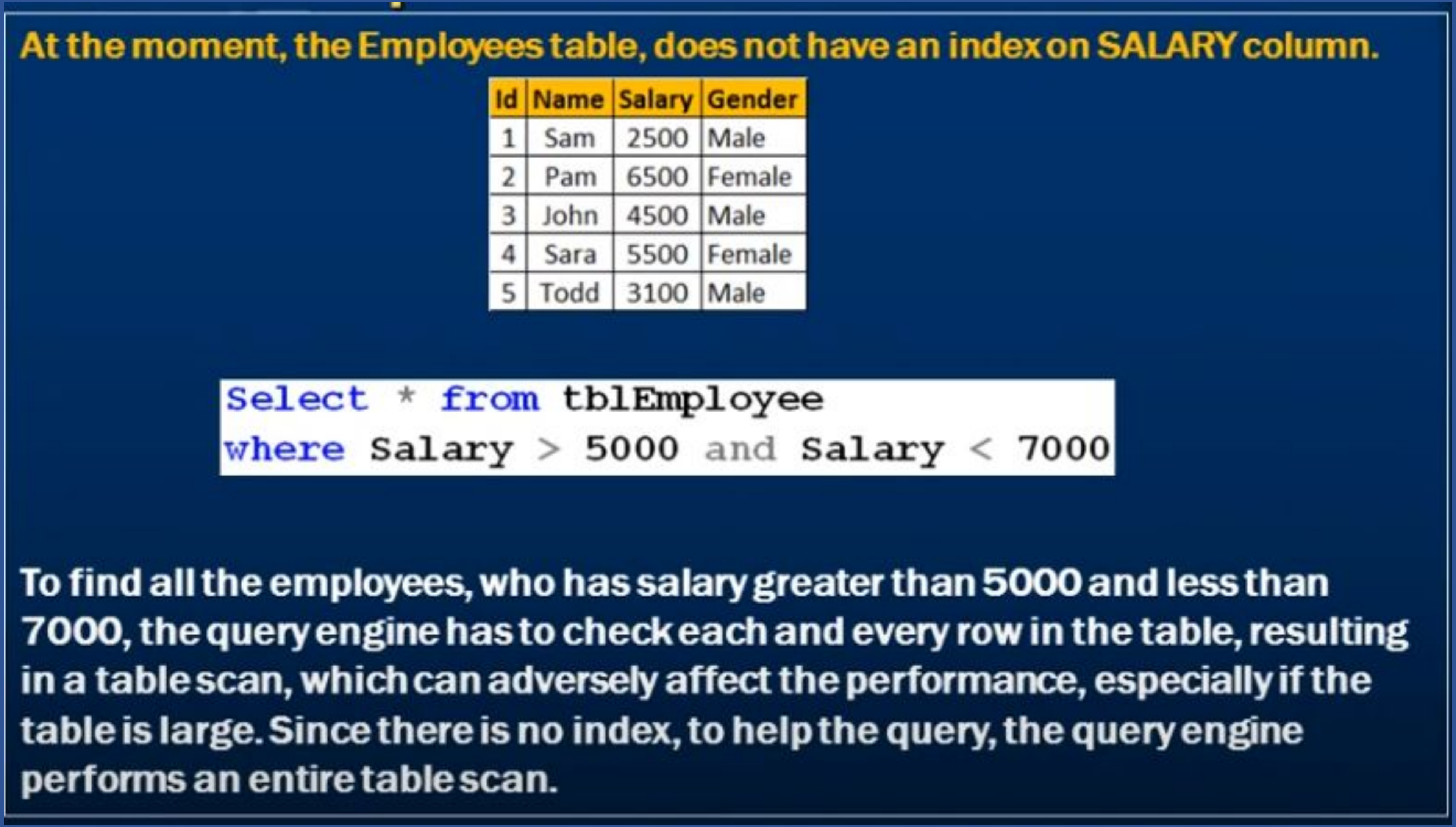


Table of Contents

To reference all pages in a book that address a particular subject, you go to the index first, which lists all the topics alphabetically, and then you go to one or more specific page numbers.

# INDEXES IN SQL

Grow Data Skills



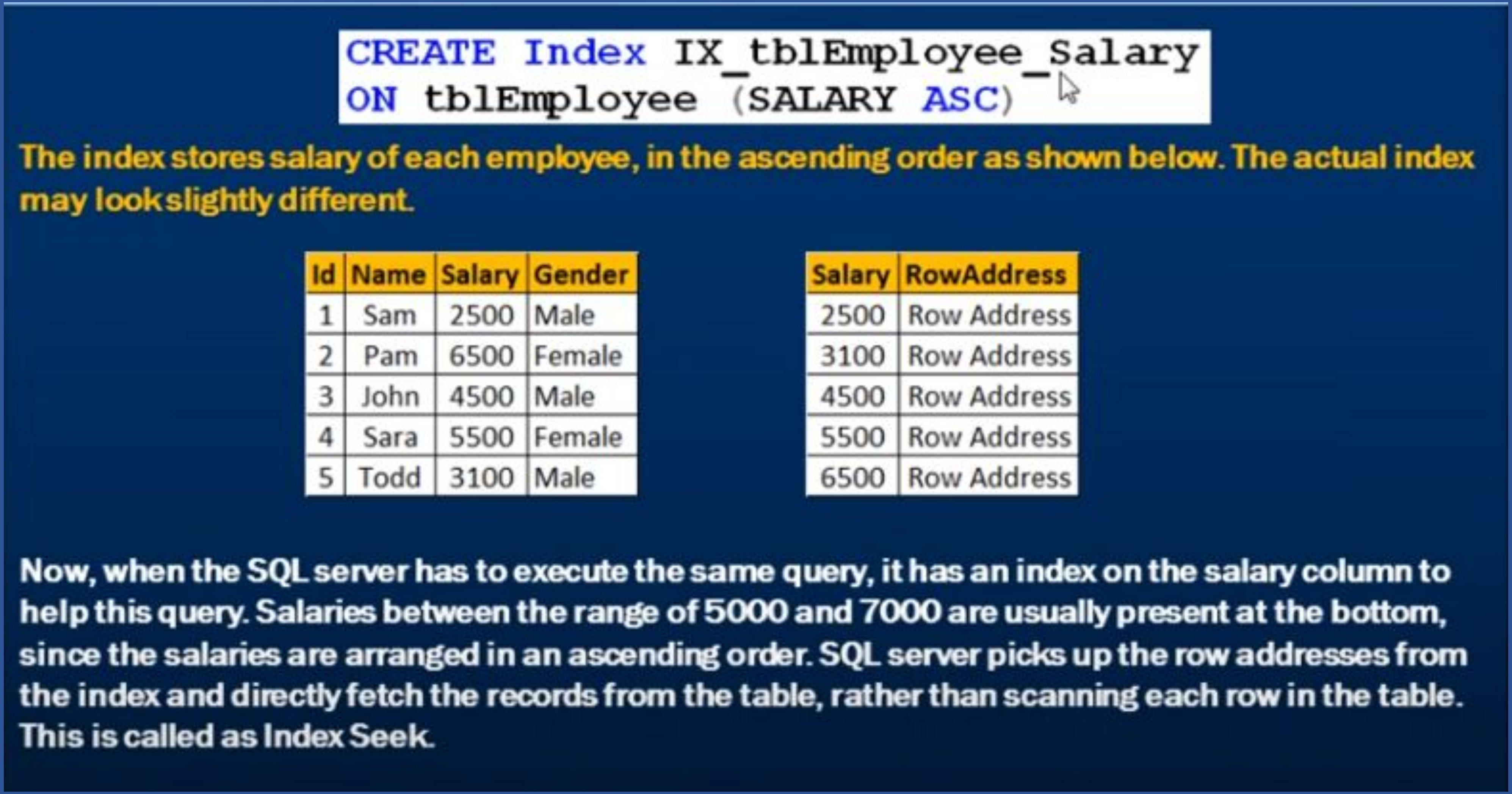
At the moment, the Employees table, does not have an index on SALARY column.

| Id | Name | Salary | Gender |
|---|---|---|---|
| 1 | Sam | 2500 | Male |
| 2 | Pam | 6500 | Female |
| 3 | John | 4500 | Male |
| 4 | Sara | 5500 | Female |
| 5 | Todd | 3100 | Male |

```sql
Select * from tblEmployee
where Salary > 5000 and Salary < 7000
```

To find all the employees, who has salary greater than 5000 and less than 7000, the query engine has to check each and every row in the table, resulting in a table scan, which can adversely affect the performance, especially if the table is large. Since there is no index, to help the query, the query engine performs an entire table scan.

# INDEXES IN SQL



```
CREATE Index IX_tblEmployee_Salary
ON tblEmployee (SALARY ASC)
```

The index stores salary of each employee, in the ascending order as shown below. The actual index may look slightly different.

| Id | Name | Salary | Gender |
|---|---|---|---|
| 1 | Sam | 2500 | Male |
| 2 | Pam | 6500 | Female |
| 3 | John | 4500 | Male |
| 4 | Sara | 5500 | Female |
| 5 | Todd | 3100 | Male |

| Salary | RowAddress |
|---|---|
| 2500 | Row Address |
| 3100 | Row Address |
| 4500 | Row Address |
| 5500 | Row Address |
| 6500 | Row Address |

Now, when the SQL server has to execute the same query, it has an index on the salary column to help this query. Salaries between the range of 5000 and 7000 are usually present at the bottom, since the salaries are arranged in an ascending order. SQL server picks up the row addresses from the index and directly fetch the records from the table, rather than scanning each row in the table. This is called as Index Seek.

# TYPES OF INDEXES

Grow Data Skills

**1. Single-Column Index:**

```
-- Create a single-column index on customer_id
CREATE INDEX idx_customer_id ON customer_orders(customer_id);
```

**2. Composite Index:**

```
-- Create a composite index on customer_id and order_date
CREATE INDEX idx_customer_order_date ON customer_orders(customer_id, order_date);
```

**3. Unique Index:**

```
-- Create a unique index on order_id
CREATE UNIQUE INDEX idx_unique_order_id ON customer_orders(order_id);
```