8.Indexes in MS-SQL Server.

Indexes are special data structures associated with tables or views that help speed up the query. SQL Server provides two types of indexes:

clustered index and non-clustered index.

8.1 Introduction to SQL Server clustered indexes

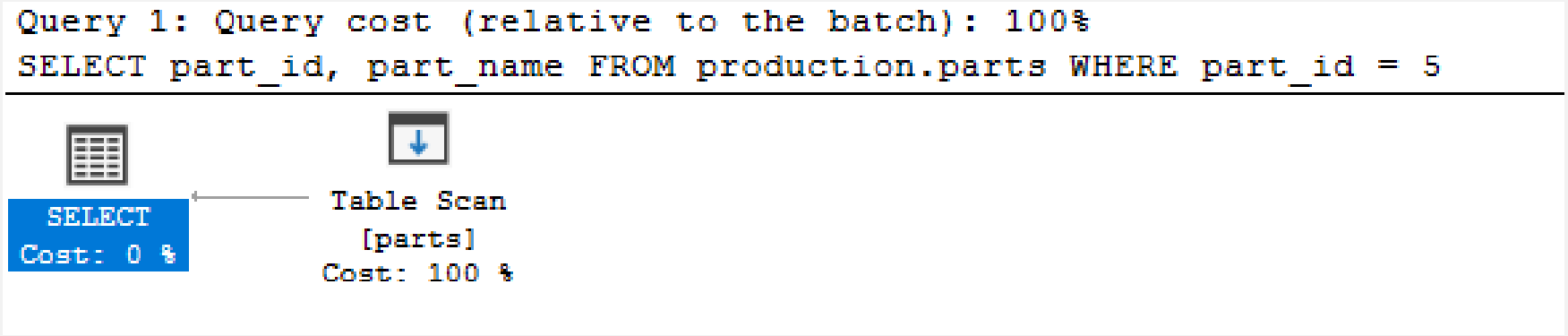
```
CREATE TABLE production.parts(
    part_id   INT NOT NULL,
    part_name VARCHAR(100)
);
```

```
INSERT INTO
    production.parts(part_id, part_name)
VALUES
    (1,'Frame'),
    (2,'Head Tube'),
    (3,'Handlebar Grip'),
    (4,'Shock Absorber'),
    (5,'Fork');
```

For example, this statement finds the part whose part id is 5.

```
1 SELECT
2     part_id,
3     part_name
4 FROM
5     production.parts
6 WHERE
7     part_id = 5;
```

If you display the estimated execution plan in SQL Server Management Studio, you can see how SQL Server came up with the following query plan:

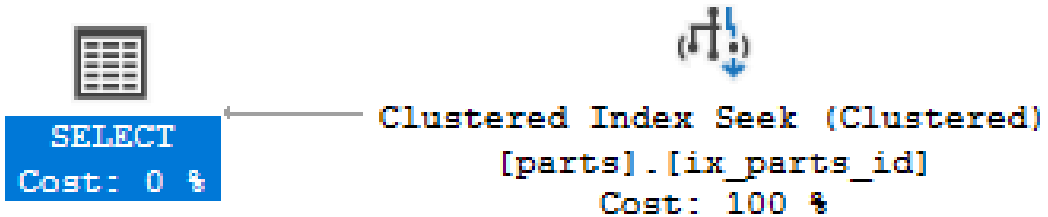


Using SQL Server CREATE CLUSTERED INDEX statement to create a clustered index. In case a table does not have a primary key, which is very rare, you can use the CREATE CLUSTERED INDEX statement to define a clustered index for the table.

```
1 CREATE CLUSTERED INDEX ix_parts_id
2 ON production.parts (part_id);
```

```
1 SELECT
2     part_id,
3     part_name
4 FROM
5     production.parts
6 WHERE
7     part_id = 5;
```

```
Query 1: Query cost (relative to the batch): 100%
SELECT part_id, part_name FROM production.parts WHERE part_id = 5
```



Introduction to SQL Server non-clustered indexes:

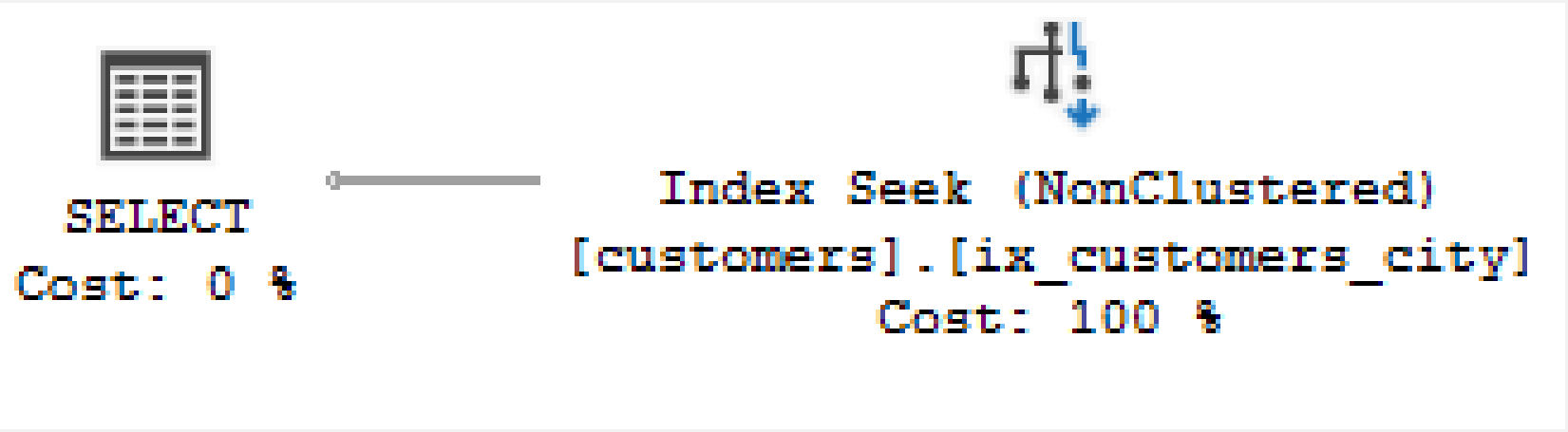
A nonclustered index is a data structure that improves the speed of data retrieval from tables. Unlike a clustered index, a nonclustered index sorts and stores data separately from the data rows in the table. It is a copy of selected columns of data from a table with the links to the associated table.

Similar to a clustered index, a nonclustered index uses the B-tree structure to organize its data.

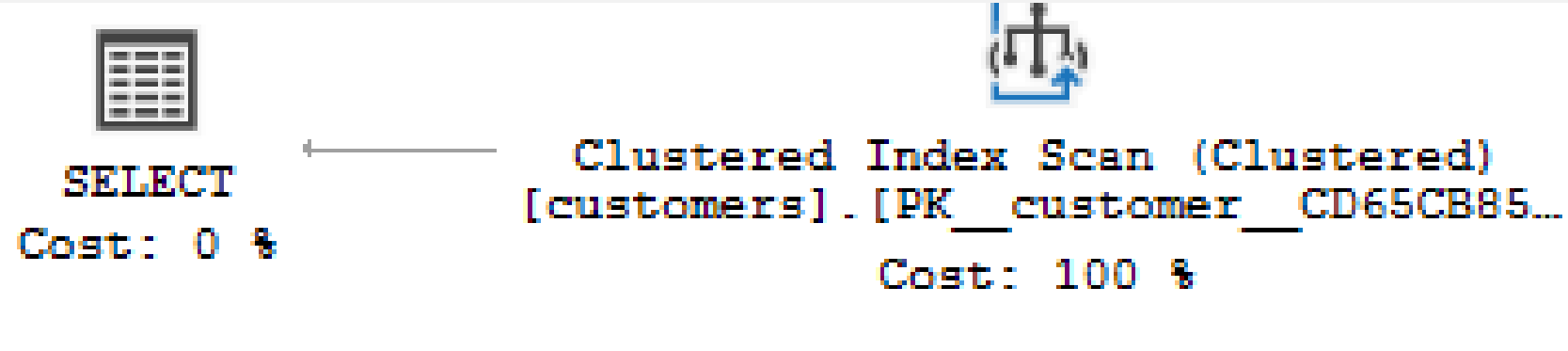
```
CREATE [NONCLUSTERED] INDEX index_name
ON table_name(column_list);
```

```
CREATE INDEX ix_customers_city
ON sales.customers(city);
```

```sql
SELECT
    customer_id,
    city
FROM
    sales.customers
WHERE
    city = 'Atwater';
```



```sql
SELECT
    customer_id,
    first_name,
    last_name
FROM
    sales.customers
WHERE
    last_name = 'Berg' AND
    first_name = 'Monika';
```



Renaming an index using the system stored procedure sp_rename

```
EXEC sp_rename
        @objname = N'sales.customers.ix_customers_city',
        @newname = N'ix_cust_city' ,
        @objtype = N'INDEX';
```

SQL Server Disable Index statements

```
ALTER INDEX ix_cust_city
ON sales.customers
DISABLE;
```

SQL Server DROP INDEX statement overview

```
DROP INDEX
    ix_cust_city ON sales.customers;

```

Clustered Index

- A Table can have ONLY 1 Clustered Index.
- A Clustered Index always has Index Id of 0.
- A Primary Key constraint creates a Clustered Index by default.
- Clustered Index enforces a logical order on the rows. Rows are ordered based on Clustering Key.
- Faster to read than non clustered as data is physically stored in index order.

Nonclustered Index

- Prior to SQL Server 2008 only 249 Nonclustered Indexes can be created. With SQL Server 2008 and above 999 Nonclustered Indexes can be created.
- Nonclustered Indexes have Index Id > 0.
- A Unique Key constraint created a Nonclustered Index by default.
- Nonclustered Index does not order actual data, It only orders columns present in the
- A table may not have any Nonclustered Indexes.
- Quicker for insert and update operations than a clustered index.