An index in a table is nothing but a pointer to its data. These are used to speed up the data retrieval from a table.

If we use indexes, the INSERT and UPDATE statements get executed in a slower phase. Whereas SELECT and WHERE get executed with in lesser time.

Creating an Index

The CREATE INDEX statement is used for creating a new Index in a table in Derby database.

Syntax

Following is the syntax of the CREATE INDEX statement −

CTREATE INDEX index\_name on table\_name (column\_name);

Example

Suppose we have created a table named Employees in Apache Derby as shown below.

CREATE TABLE Emp ( Id INT NOT NULL GENERATED ALWAYS AS IDENTITY,

Name VARCHAR(255),

Salary INT NOT NULL,

Location VARCHAR(255),

Phone\_Number BIGINT

);

The following SQL statement creates an index on the column named Salary in the table Employees.

ij> CREATE INDEX example\_index on Emp (Salary);

0 rows inserted/updated/deleted

Creating a UNIQUE index

In Apache Derby, UNIQUE indexes are used for data integration. Once you create a UNIQUE index on a column in a table, it does not allow duplicate values.

Syntax

Following is the syntax of creating a unique index.

CREATE UNIQUE INDEX index\_name on table\_name (column\_name);

Example

Following example creates a UNIQUE index on the column Id of the table Employee.

ij> CREATE UNIQUE INDEX unique\_index on Emp (Phone\_Number);

0 rows inserted/updated/deleted

Once you have created a unique index on a column, you cannot enter same values for that column in another row. In short, a column which is has a UNIQE index will not allow duplicate values.

Insert a row in the Emp table as shown below

ij> INSERT INTO Emp(Name, Salary, Location, Phone\_Number) VALUES ('Amit',

45000, 'Hyderabad', 9848022338);

1 row inserted/updated/deleted

Since we have created a unique index on the column Phone\_No, if you ty to enter the same value as in the previous record, it shows an error.

ij> INSERT INTO Emp(Name, Salary, Location, Phone\_Number) VALUES ('Sumit',

35000, 'Chennai', 9848022338);

ERROR 23505: The statement was aborted because it would have caused a duplicate

key value in a unique or primary key constraint or unique index identified by

'UNIQUE\_INDEX' defined on 'EMP'.

Creating a COMPOSITE index

You can create a single index on two rows and it is called Composite index.

Syntax

Following is the syntax of the composite index.

CREATE INDEX index\_name on table\_name (column\_name1, column\_name2);

Example

Following index creates a composite index on the columns Name and Location.

ij> CREATE INDEX composite\_index on Emp (Name, Location);

0 rows inserted/updated/deleted

Displaying the Indexes

The SHOW INDEXES query displays the list of indexes on a table.

Syntax

Following is the syntax of the SHOW INDEXES statement −

SHOW INDEXES FROM table\_name;

Example

Following example, i displays the indexes on the table Employees.

ij> SHOW INDEXES FROM Emp;

This produces the following result.

ij> SHOW INDEXES FROM Emp;

TABLE\_NAME |COLUMN\_NAME |NON\_U&|TYPE|ASC&|CARDINA&|PAGES

----------------------------------------------------------------------------

EMP |PHONE\_NUMBER |false |3 |A |NULL |NULL

EMP |NAME |true |3 |A |NULL |NULL

EMP |LOCATION |true |3 |A |NULL |NULL

EMP |SALARY |true |3 |A |NULL |NULL

4 rows selected

Dropping Indexes

The Drop Index statement deletes/drops the given index on a column.

Syntax

Following is the syntax of the DROP INDEX statement.

DROP INDEX index\_name;

Example

Following example drops an indexes named composite\_index and unique\_index created above.

ij> DROP INDEX composite\_index;

0 rows inserted/updated/deleted

ij>Drop INDEX unique\_index;

0 rows inserted/updated/deleted

Now, if you verify the list of indexes you can see index on one column since we have deleted the remaining.

ij> SHOW INDEXES FROM Emp;

TABLE\_NAME |COLUMN\_NAME |NON\_U&|TYPE|ASC&|CARDINA&|PAGES

----------------------------------------------------------------------------

EMP |SALARY |true |3 |A |NULL |NULL

1 row selected

Handling Indexes using JDBC program

Following JDBC program demonstrates how to create drop indexes on a column in a table.

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.sql.Statement;

public class IndexesExample {

public static void main(String args[]) throws Exception {

//Registering the driver

Class.forName("org.apache.derby.jdbc.EmbeddedDriver");

//Getting the Connection object

String URL = "jdbc:derby:MYDATABASE;create=true";

Connection conn = DriverManager.getConnection(URL);

//Creating the Statement object

Statement stmt = conn.createStatement();

//Creating the Emp table

String createQuery = "CREATE TABLE Emp( "

+ "Id INT NOT NULL GENERATED ALWAYS AS IDENTITY, "

+ "Name VARCHAR(255), "

+ "Salary INT NOT NULL, "

+ "Location VARCHAR(255), "

+ "Phone\_Number BIGINT )";

stmt.execute(createQuery);

System.out.println("Table created");

System.out.println(" ");

//Creating an Index on the column Salary

stmt.execute("CREATE INDEX example\_index on Emp (Salary)");

System.out.println("Index example\_index inserted");

System.out.println(" ");

//Creating an Unique index on the column Phone\_Number

stmt.execute("CREATE UNIQUE INDEX unique\_index on Emp (Phone\_Number)");

System.out.println("Index unique\_index inserted");

System.out.println(" ");

//Creating a Composite Index on the columns Name and Location

stmt.execute("CREATE INDEX composite\_index on Emp (Name, Location)");

System.out.println("Index composite\_index inserted");

System.out.println(" ");

//listing all the indexes

System.out.println("Listing all the columns with indexes");

//Dropping indexes

System.out.println("Dropping indexes unique\_index and, composite\_index ");

stmt.execute("Drop INDEX unique\_index");

stmt.execute("DROP INDEX composite\_index");

}

}

Output

On executing, this generates the following result

Table created

Index example\_index inserted

Index unique\_index inserted

Index composite\_index inserted

Listing all the columns with indexes

Dropping indexes unique\_index and, composite\_index