The ALTER TABLE statement, allows you to alter an existing table. Using this you can do the following −

* Add a column, add a constraint
* Drop a column, drop a constraint
* Change the row level locking of a table

Let us assume we have created a table named Employees as shown below −

ij> CREATE TABLE Employees (

Id INT NOT NULL GENERATED ALWAYS AS IDENTITY,

Name VARCHAR(255),

Salary INT NOT NULL,

Location VARCHAR(255),

PRIMARY KEY (Id)

);

And, inserted four records using the insert statement as −

ij> INSERT INTO Employees (Name, Salary, Location) VALUES

('Amit', 30000, 'Hyderabad'),

('Kalyan', 40000, 'Vishakhapatnam'),

('Renuka', 50000, 'Delhi'),

('Archana', 15000, 'Mumbai');

Adding a column to a Table

Following is the syntax to add a column to a table using ALTER statement.

ALTER TABLE table\_name ADD COLUMN column\_name column\_type;

Example

Using ALTER statement, we are trying to add a new column named Age with the type integer.

ALTER TABLE Employees ADD COLUMN Age INT;

0 rows inserted/updated/deleted

Add another column named Phone\_No with the type integer.

ALTER TABLE Employees ADD COLUMN Phone\_No BIGINT;

0 rows inserted/updated/deleted

The DESCRIBE command describes specified table by listing the columns and their details, if the table exists. If you DESCRIBE, the table Employees you can observe the newly added columns as shown below −

ij> DESCRIBE Employees;

COLUMN\_NAME |TYPE\_NAME|DEC&|NUM&|COLUM&|COLUMN\_DEF|CHAR\_OCTE&|IS\_NULL&

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

ID |INTEGER |0 |10 |10 |AUTOINCRE&|NULL |NO

NAME |VARCHAR |NULL|NULL|255 |NULL |510 |YES

SALARY |INTEGER |0 |10 |10 |NULL |NULL |NO

LOCATION |VARCHAR |NULL|NULL|255 |NULL |510 |YES

AGE |INTEGER |0 |10 |10 |NULL |NULL |YES

PHONE\_NO |INTEGER |0 |10 |10 |NULL |NULL |YES

6 rows selected

Adding a constraint to a table

Following is the syntax to add a constraint to a column of a table using ALTER statement.

ALTER TABLE table\_name ADD CONSTRAINT constraint\_name constraint (column\_name);

Where **constraint** can be NOT NULL, NULL, PRIMARY KEY, UNIQUE, FOREIGN KEY, CHECK.

Example

Using ALTER statement, we are trying to add constraint **UNIQUE** to the Phone\_No column.

ij> ALTER TABLE Employees ADD CONSTRAINT New\_Constraint UNIQUE(Phone\_No);

0 rows inserted/updated/deleted

Once, you add a UNIQUE constraint to a column, it cannot have the same values for two rows, i.e., phone number should be unique for each employee.

If you try to add two columns with a same phone number, you will get an exception as shown below.

ij> INSERT INTO Employees (Name, Salary, Location, Age, Phone\_No) VALUES

('Amit', 30000, 'Hyderabad', 30, 9848022338);

1 row inserted/updated/deleted

ij> INSERT INTO Employees (Name, Salary, Location, Age, Phone\_No) VALUES

('Sumit', 35000, 'Chennai', 25, 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

'NEW\_CONSTRAINT' defined on 'EMPLOYEES'.

Dropping a constraint from a table

Following is the syntax to drop a constraint of a column −

ALTER TABLE table\_name DROP CONSTRAINT constraint\_name;

Example

The following query deletes the constraint name New\_Constraint on the column Phone\_No created above.

ij> ALTER TABLE Employees DROP CONSTRAINT New\_Constraint;

0 rows inserted/updated/deleted

Since we have removed the UNIQUE constraint on the column Phone\_No, you can add columns with the same phone number.

ij> INSERT INTO Employees (Name, Salary, Location, Age, Phone\_No) VALUES

('Sumit', 35000, 'Chennai', 25, 9848022338);

1 row inserted/updated/deleted

You can verify the contents of the table ij> select \* from Employees as follows −

ID |NAME |SALARY |LOCATION |AGE |PHONE\_NO

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

1 |Amit |30000 |Hyderabad |30 |9848022338

2 |Sumit |35000 |Chennai |25 |9848022338

2 rows selected

Dropping a column from a table

Following is the syntax to drop a column of a column.

ALTER TABLE table\_name DROP COLUMN column\_name;

Example

Following query deletes the column named **age of the employee** −

ij> ALTER TABLE Employees DROP COLUMN Age;

0 rows inserted/updated/deleted

If you describe the table, you can see only 4 columns.

ij> DESCRIBE Employees;

COLUMN\_NAME |TYPE\_NAME|DEC&|NUM&|COLUM&|COLUMN\_DEF|CHAR\_OCTE&|IS\_NULL&

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

ID |INTEGER |0 |10 |10 |AUTOINCRE&|NULL |NO

NAME |VARCHAR |NULL|NULL|255 |NULL |510 |YES

SALARY |INTEGER |0 |10 |10 |NULL |NULL |NO

LOCATION |VARCHAR |NULL|NULL|255 |NULL |510 |YES

PHONE\_NO |BIGINT |0 |10 |19 |NULL |NULL |YES

Altering table using JDBC program

Following is the JDBC program to alter a table using the ALTER query −

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

public class AlterTableExample {

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:sampleDB;create=true";

Connection conn = DriverManager.getConnection(URL);

//Creating the Statement object

Statement stmt = conn.createStatement();

//Executing the query

String createQuery = "CREATE TABLE Employees( "

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

+ "Name VARCHAR(255), "

+ "Salary INT NOT NULL, "

+ "Location VARCHAR(255), "

+ "PRIMARY KEY (Id))";

stmt.execute(createQuery);

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

System.out.println(" ");

//Executing the query

String insertQuery = "INSERT INTO Employees("

+ "Name, Salary, Location) VALUES "

+ "('Amit', 30000, 'Hyderabad'), "

+ "('Kalyan', 40000, 'Vishakhapatnam'), "

+ "('Renuka', 50000, 'Delhi'), "

+ "('Archana', 15000, 'Mumbai'), "

+ "('Trupti', 45000, 'Kochin')";

stmt.execute(insertQuery);

System.out.println("Values inserted");

System.out.println(" ");

//Executing the query

String selectQuery = "SELECT \* FROM Employees";

ResultSet rs = stmt.executeQuery(selectQuery);

System.out.println("Contents of the table after inserting the table");

while(rs.next()) {

System.out.println("Id: "+rs.getString("Id"));

System.out.println("Name: "+rs.getString("Name"));

System.out.println("Salary: "+rs.getString("Salary"));

System.out.println("Location: "+rs.getString("Location"));

}

System.out.println(" ");

//Altering the table

stmt.execute("ALTER TABLE Employees ADD COLUMN Age INT");

stmt.execute("ALTER TABLE Employees ADD COLUMN Phone\_No BigINT");

stmt.execute("ALTER TABLE Employees " + "ADD CONSTRAINT New\_Constraint UNIQUE(Phone\_No)");

stmt.execute("INSERT INTO Employees "

+ "(Name, Salary, Location, Age, Phone\_No) "

+ "VALUES ('Amit', 30000, 'Hyderabad', 30, 9848022338)");

ResultSet alterResult = stmt.executeQuery("Select \* from Employees");

System.out.println("Contents of the table after altering "

+ "the table and inserting values to it: ");

while(alterResult.next()) {

System.out.println("Id: "+alterResult.getString("Id"));

System.out.println("Name: "+alterResult.getString("Name"));

System.out.println("Salary: "+alterResult.getString("Salary"));

System.out.println("Location: "+alterResult.getString("Location"));

System.out.println("Age: "+alterResult.getString("Age"));

System.out.println("Phone\_No: "+alterResult.getString("Phone\_No"));

}

}

}

Output

On executing the above program, the following output will be generated −

Table created

Values inserted

Contents of the table after inserting the table

Id: 1

Name: Amit

Salary: 30000

Location: Hyderabad

Id: 2

Name: Kalyan

Salary: 40000

Location: Vishakhapatnam

Id: 3

Name: Renuka

Salary: 50000

Location: Delhi

Id: 4

Name: Archana

Salary: 15000

Location: Mumbai

Id: 5

Name: Trupti

Salary: 45000

Location: Kochin

Contents of the table after altering the table and inserting values to it:

Id: 1

Name: Amit

Salary: 30000

Location: Hyderabad

Age: null

Phone\_No: null

Id: 2

Name: Kalyan

Salary: 40000

Location: Vishakhapatnam

Age: null

Phone\_No: null

Id: 3

Name: Renuka

Salary: 50000

Location: Delhi

Age: null

Phone\_No: null

Id: 4

Name: Archana

Salary: 15000

Location: Mumbai

Age: null

Phone\_No: null

Id: 5

Name: Trupti

Salary: 45000

Location: Kochin

Age: null

Phone\_No: null

Id: 6

Name: Amit

Salary: 30000

Location: Hyderabad

Age: 30

Phone\_No: 9848022338