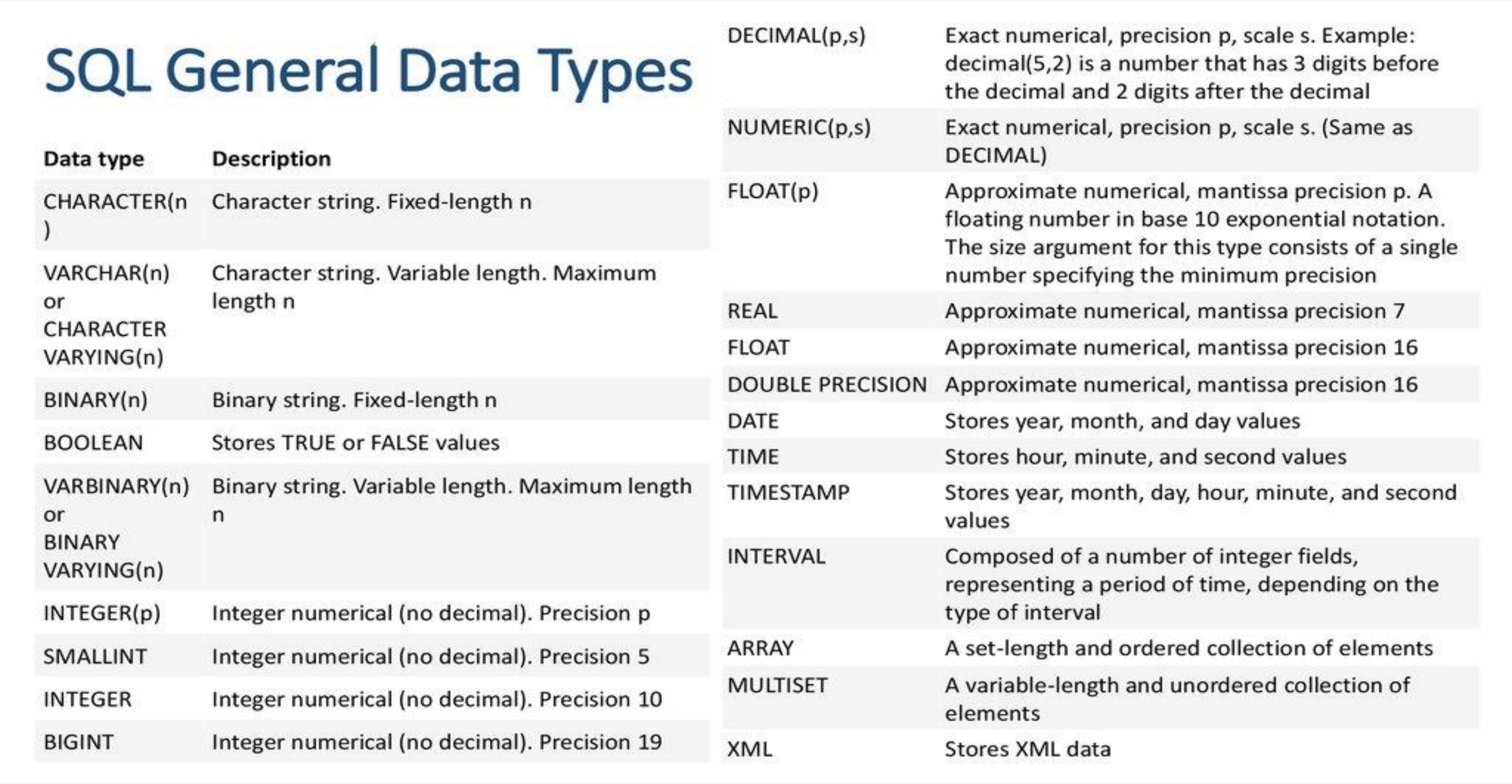
SQL

CREATE TABLE

COMMAND

DATA TYPES IN SQL

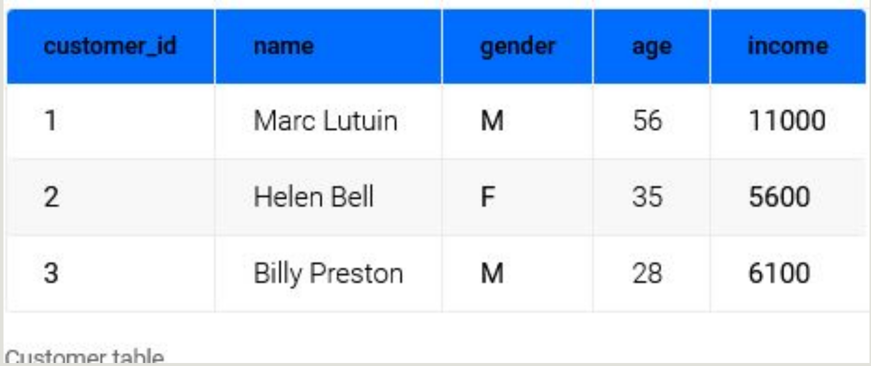


SQL General Data Types

Data type	Description
CHARACTER(n)	Character string. Fixed-length n
VARCHAR(n) or CHARACTER VARYING(n)	Character string. Variable length. Maximum length n
BINARY(n)	Binary string. Fixed-length n
BOOLEAN	Stores TRUE or FALSE values
VARBINARY(n) or BINARY VARYING(n)	Binary string. Variable length. Maximum length n
INTEGER(p)	Integer numerical (no decimal). Precision p
SMALLINT	Integer numerical (no decimal). Precision 5
INTEGER	Integer numerical (no decimal). Precision 10
BIGINT	Integer numerical (no decimal). Precision 19
DECIMAL(p,s)	Exact numerical, precision p, scale s. Example: decimal(5,2) is a number that has 3 digits before the decimal and 2 digits after the decimal
NUMERIC(p,s)	Exact numerical, precision p, scale s. (Same as DECIMAL)
FLOAT(p)	Approximate numerical, mantissa precision p. A floating number in base 10 exponential notation. The size argument for this type consists of a single number specifying the minimum precision
REAL	Approximate numerical, mantissa precision 7
FLOAT	Approximate numerical, mantissa precision 16
DOUBLE PRECISION	Approximate numerical, mantissa precision 16
DATE	Stores year, month, and day values
TIME	Stores hour, minute, and second values
TIMESTAMP	Stores year, month, day, hour, minute, and second values
INTERVAL	Composed of a number of integer fields, representing a period of time, depending on the type of interval
ARRAY	A set-length and ordered collection of elements
MULTISET	A variable-length and unordered collection of elements
XML	Stores XML data

What Is a Database Table?

- Like an Excel spreadsheet, a database table organizes and stores information using rows and columns. Inside the table, each data item is defined with one row (record) and each attribute of the item is stored within a column (cell).



customer_id	name	gender	age	income
1	Marc Lutuin	M	56	11000
2	Helen Bell	F	35	5600
3	Billy Preston	M	28	6100

Customer table

- This table contains three rows. Each row represents one person (Marc, Helen, and Billy). There are five columns in this table (id, name, gender, age and income) and **each column represents one customer attribute.**

What Is a Relational Database?

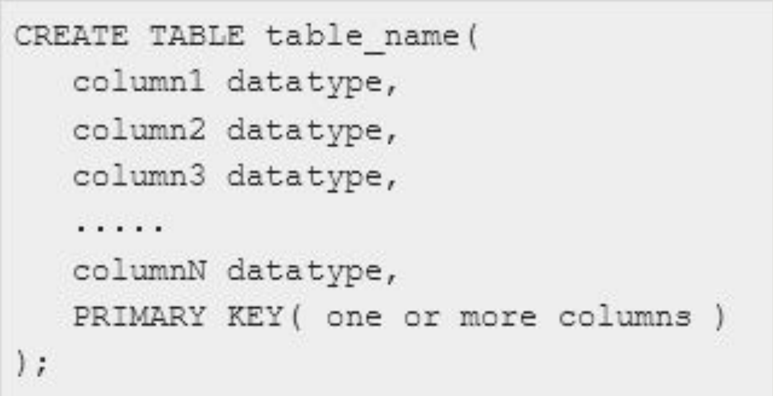
- A **relational database** can have many tables; each table will contain data that’s related in some way.
- For example, a bank could have:
  - The **customer** table shown above.
  - A **loan** table that stores information about customer loans.
  - A **balance** table with data about customers’ account balances.
- All three tables are related and can be in the same database. You can create other tables as well, depending on your needs.

CREATE TABLE -INTRO

- The CREATE TABLE is a **DDL statement** which is used to create tables in the database.
- The CREATE TABLE command is used to specify a new relation **by giving it a name and specifying its attributes and initial constraints.**
- The attributes are specified first, and each attribute is given a name, a data type to specify its domain of values, and any attribute constraints, such as NOT NULL.
- The key, entity integrity, and referential integrity constraints can be specified within the CREATE TABLE statement after the attributes are declared, or they can be added later using the ALTER TABLE command.

SYNTAX

- If you want to create a table, you should name the table and define its column and each column's data type.
- The basic syntax of the CREATE TABLE statement is as follows:

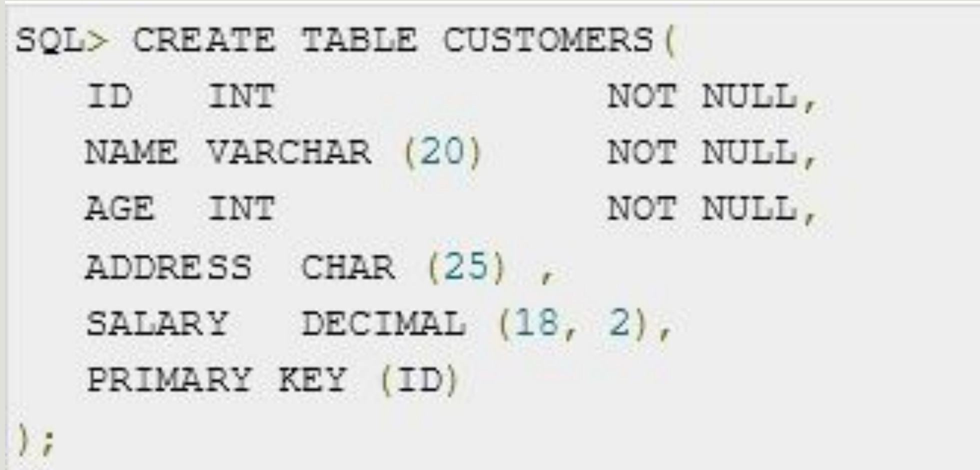


```
CREATE TABLE table_name(
   column1 datatype,
   column2 datatype,
   column3 datatype,
   .....
   columnN datatype,
   PRIMARY KEY( one or more columns )
);
```

Example

Example

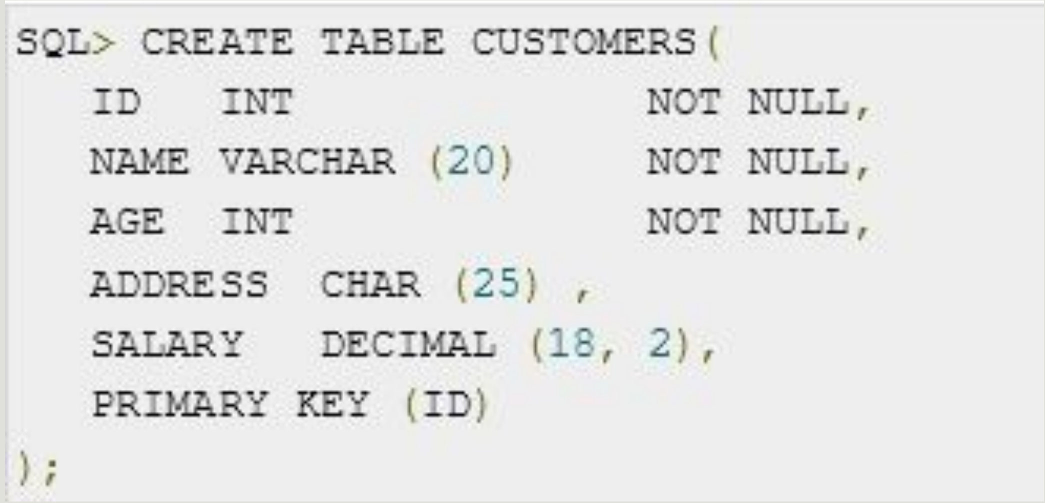
The following code block is an example, which creates a CUSTOMERS table with an ID as a primary key and NOT NULL are the constraints showing that these fields cannot be NULL while creating records in this table:



```
SQL> CREATE TABLE CUSTOMERS(
   ID   INT              NOT NULL,
   NAME VARCHAR (20)     NOT NULL,
   AGE  INT              NOT NULL,
   ADDRESS  CHAR (25) ,
   SALARY   DECIMAL (18, 2),
   PRIMARY KEY (ID)
);
```

Example

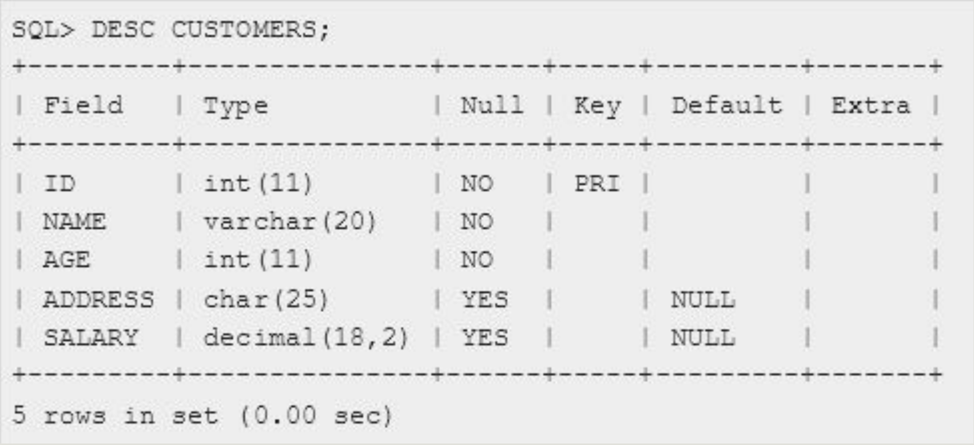
The following code block is an example, which creates a CUSTOMERS table with an ID as a primary key and NOT NULL are the constraints showing that these fields cannot be NULL while creating records in this table:



```
SQL> CREATE TABLE CUSTOMERS(
   ID   INT                 NOT NULL,
   NAME VARCHAR (20)        NOT NULL,
   AGE  INT                 NOT NULL,
   ADDRESS  CHAR (25) ,
   SALARY   DECIMAL (18, 2),
   PRIMARY KEY (ID)
);
```

DESC Command

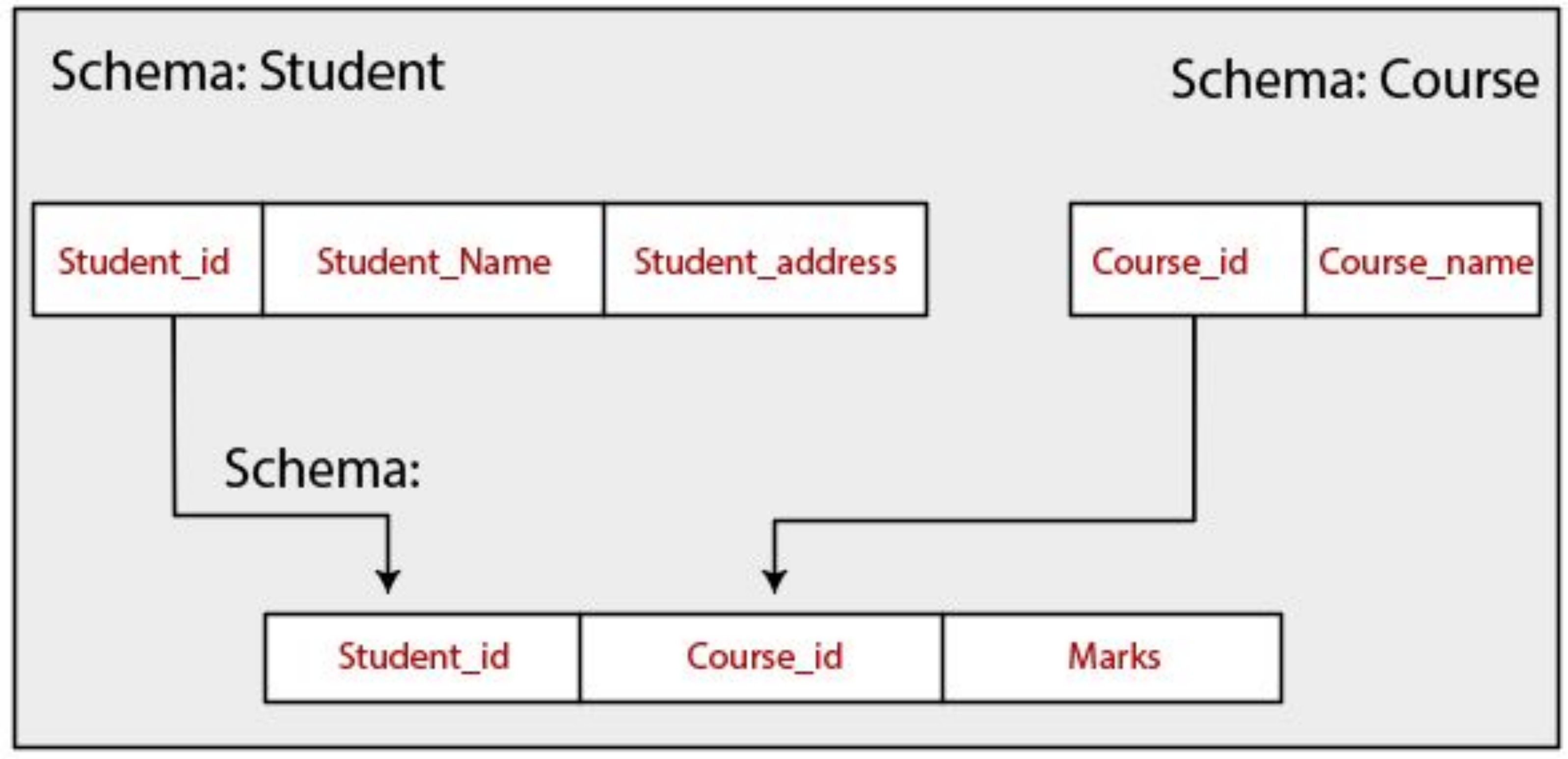
You can verify if your table has been created successfully by looking at the message displayed by the SQL server, otherwise you can use the **DESC** command as follows:



```
SQL> DESC CUSTOMERS;
+---------+---------------+------+-----+---------+-------+
| Field   | Type          | Null | Key | Default | Extra |
+---------+---------------+------+-----+---------+-------+
| ID      | int(11)       | NO   | PRI |         |       |
| NAME    | varchar(20)   | NO   |     |         |       |
| AGE     | int(11)       | NO   |     |         |       |
| ADDRESS | char(25)      | YES  |     | NULL    |       |
| SALARY  | decimal(18,2) | YES  |     | NULL    |       |
+---------+---------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

Now, you have CUSTOMERS table available in your database which you can use to store the required information related to customers.

Example



Schema: Student

Student_id	Student_Name	Student_address

Schema: Course

Course_id	Course_name

Schema:

Student_id	Course_id	Marks

INSERT STATEMENT

- The SQL **INSERT INTO** Statement is used to add new rows of data to a table in the database.

- There are two ways of using INSERT INTO statement for inserting rows:

1) Only values:

- This method is to specify only the value of data to be inserted without the column names.

- **If you are adding values for all the columns of the table**, you do not need to specify the column names in the SQL query. However, make sure **the order of the values is in the same order as the columns in the table.** Here, the **INSERT INTO syntax** would be as follows:

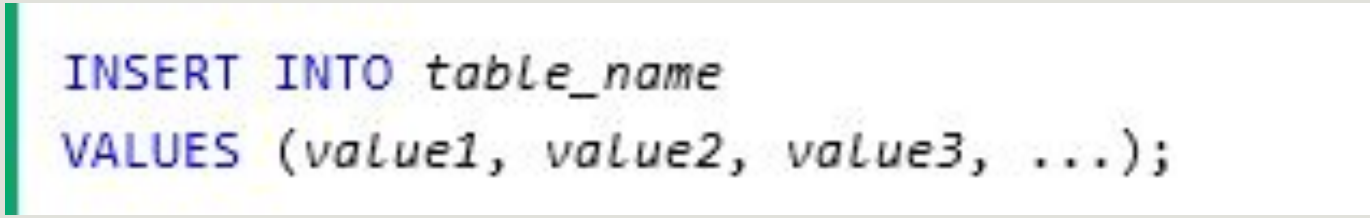
INSERT STATEMENT

- The SQL **INSERT INTO** Statement is used to add new rows of data to a table in the database.
- There are two ways of using INSERT INTO statement for inserting rows:

1) Only values:

- This method is to specify only the value of data to be inserted without the column names.
- **If you are adding values for all the columns of the table**, you do not need to specify the column names in the SQL query. However, make sure **the order of the values is in the same order as the columns in the table.** Here, the **INSERT INTO syntax** would be as follows:

INSERT STATEMENT



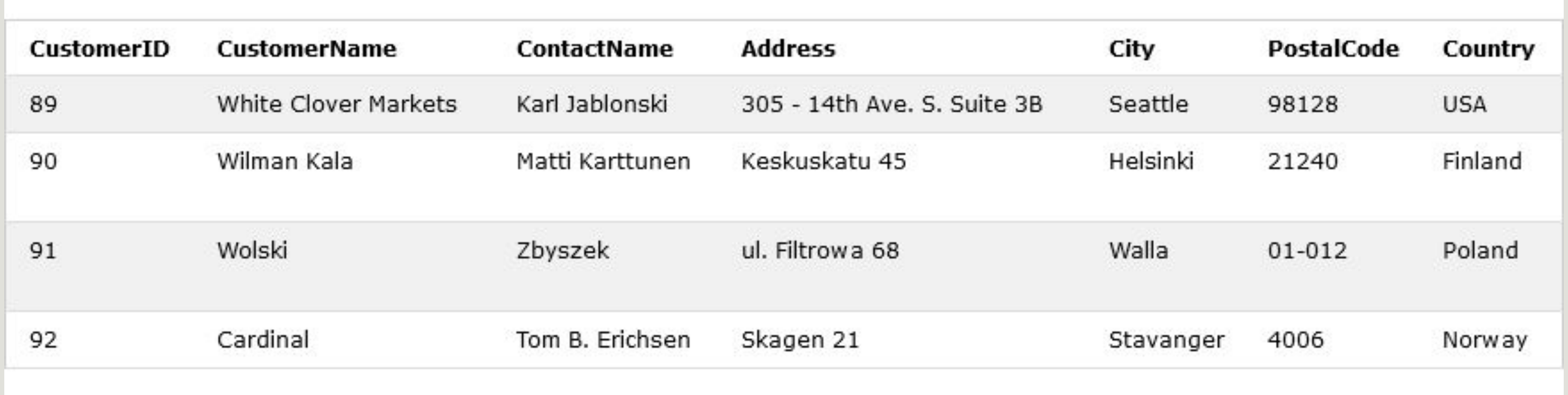
```
INSERT INTO table_name
VALUES (value1, value2, value3, ...);
```

- The following SQL statement inserts a new record in the "Customers" table:

  INSERT INTO CUSTOMER

  VALUES (92, 'Cardinal', 'Tom B. Erichsen', 'Skagen 21', 'Stavanger', '4006', 'Norway');

INSERT STATEMENT

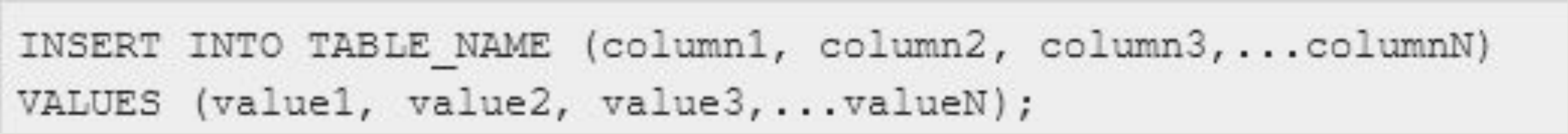


CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
89	White Clover Markets	Karl Jablonski	305 - 14th Ave. S. Suite 3B	Seattle	98128	USA
90	Wilman Kala	Matti Karttunen	Keskuskatu 45	Helsinki	21240	Finland
91	Wolski	Zbyszek	ul. Filtrowa 68	Walla	01-012	Poland
92	Cardinal	Tom B. Erichsen	Skagen 21	Stavanger	4006	Norway

INSERT STATEMENT

2) Insert Data Only in Specified Columns

- It is also possible to only insert data in specific columns.



```
INSERT INTO TABLE_NAME (column1, column2, column3,...columnN)
VALUES (value1, value2, value3,...valueN);
```

- The following SQL statement will insert a new record, but only insert data in the “Customer id”, "CustomerName", "City", and "Country" columns.

INSERT STATEMENT

```
INSERT INTO Customers (Customerid, CustomerName, City, Country)

VALUES (100,'Cardinal', 'Stavanger', 'Norway');
```