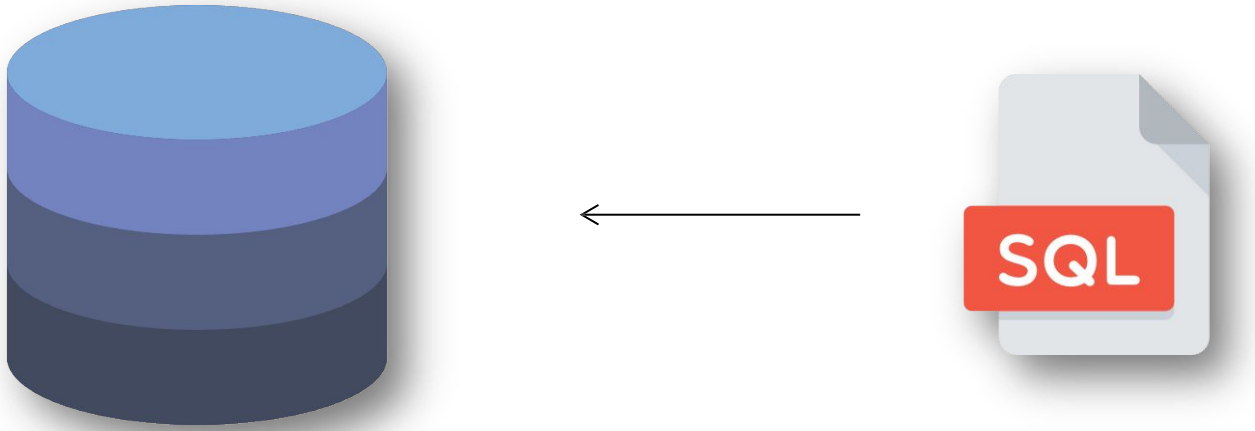
# SQL Basics

# Introduction to SQL

SQL stands for Structured Query Language which is a standard language for accessing and manipulating databases.



# Categories of SQL Commands


| Data Query Language | Data Definition Language | Data Manipulation Language | Data Control Language |
| --- | --- | --- | --- |
| Select | Create Table | Insert | Grant |
| | Alter Table | Update | Revoke |

# Tables in SQL

A table is a database object which comprises rows and columns.

| e_id | e_name | e_salary | e_age | e_gender | e_dept |
|---|---|---|---|---|---|
| 1 | Sam | 95000 | 45 | Male | Operations |
| 2 | Bob | 80000 | 21 | Male | Support |
| 3 | Anne | 125000 | 25 | Female | Analytics |
| 4 | Julia | 73000 | 30 | Female | Analytics |
| 5 | Matt | 159000 | 33 | Male | Sales |
| 6 | Jeff | 112000 | 27 | Male | Operations |

# Fields

A field provides specific information about the data in a table.

| e_id | e_name | e_salary | e_age | e_gender | e_dept |
|---|---|---|---|---|---|
| 1 | Sam | 95000 | 45 | Male | Operations |
| 2 | Bob | 80000 | 21 | Male | Support |
| 3 | Anne | 125000 | 25 | Female | Analytics |
| 4 | Julia | 73000 | 30 | Female | Analytics |
| 5 | Matt | 159000 | 33 | Male | Sales |
| 6 | Jeff | 112000 | 27 | Male | Operations |

# Records

Each individual entry in a table is called a record.

| e_id | e_name | e_salary | e_age | e_gender | e_dept |
|---|---|---|---|---|---|
| 1 | Sam | 95000 | 45 | Male | Operations |
| 2 | Bob | 80000 | 21 | Male | Support |
| 3 | Anne | 125000 | 25 | Female | Analytics |
| 4 | Julia | 73000 | 30 | Female | Analytics |
| 5 | Matt | 159000 | 33 | Male | Sales |
| 6 | Jeff | 112000 | 27 | Male | Operations |

# Fields and Records

| e_id | e_name | e_salary | e_age | e_gender | e_dept |
|---|---|---|---|---|---|
| 1 | Sam | 95000 | 45 | Male | Operations |
| 2 | Bob | 80000 | 21 | Male | Support |
| 3 | Anne | 125000 | 25 | Female | Analytics |
| 4 | Julia | 73000 | 30 | Female | Analytics |
| 5 | Matt | 159000 | 33 | Male | Sales |
| 6 | Jeff | 112000 | 27 | Male | Operations |

Record

| e_id | e_name | e_salary | e_age | e_gender | e_dept |
|---|---|---|---|---|---|
| 1 | Sam | 95000 | 45 | Male | Operations |
| 2 | Bob | 80000 | 21 | Male | Support |
| 3 | Anne | 125000 | 25 | Female | Analytics |
| 4 | Julia | 73000 | 30 | Female | Analytics |
| 5 | Matt | 159000 | 33 | Male | Sales |
| 6 | Jeff | 112000 | 27 | Male | Operations |

Field

# Creating a Database: Syntax

```
CREATE DATABASE databasename;
```

# Using a Database: Syntax

```
USE [databasename];
```

# Dropping a Database: Syntax

```
DROP DATABASE [databasename];
```

# Constraints

Rules for the data in a table can be specified using SQL constraints.The kinds of data that can be entered into a table are restricted by constraints. This validates the reliability and accuracy of the data in the table.

**Types of constraints:**

NOT NULL - prevents a column from having a NULL value.
UNIQUE - ensure that each value in a column is unique.
PRIMARY KEY - A combination of a NOT NULL and UNIQUE.
FOREIGN KEY - A field or column used to create a connection between two tables is known as a foreign key.
CHECK - check whether the values in a column satisfy a particular requirement.
DEFAULT - Sets a default value for a column in the absence of a value

# Not Null Constraint

The NOT NULL constraint enforces a column to NOT accept NULL values. (like CustomerID)

| CustomerID | CustomerName | ContactName | Address |
|---|---|---|---|
| 1 | Alfreds Futterkiste | Maria Anders | Obere Str. 57 |
| 2 | Ana Trujillo Emparedados y helados | Ana Trujillo | Avda. de la Constitución 2222 |
| 3 | Antonio Moreno Taquería | Antonio Moreno | Mataderos 2312 |
| 4 | Around the Horn | Thomas Hardy | 120 Hanover Sq. |

# Default Constraint

The DEFAULT constraint is used to set a default value for a column.

```
CREATE TABLE Persons (
    ID int NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int,
    City varchar(255) DEFAULT ‘Mathura'
);
```

# Unique Constraint

The UNIQUE constraint ensures that all values in a column are different. Both the UNIQUE and PRIMARY KEY constraints provide a guarantee for uniqueness for a column or set of columns.

```
CREATE TABLE Persons (
    ID int NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int,
    UNIQUE (ID)
);
```

# Primary Key Constraint

The PRIMARY KEY constraint uniquely identifies each record in a table.
Primary keys must contain UNIQUE values, and cannot contain NULL values.

```
CREATE TABLE Persons (
    PersonID int NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int,
    PRIMARY KEY (PersonID)
);
```

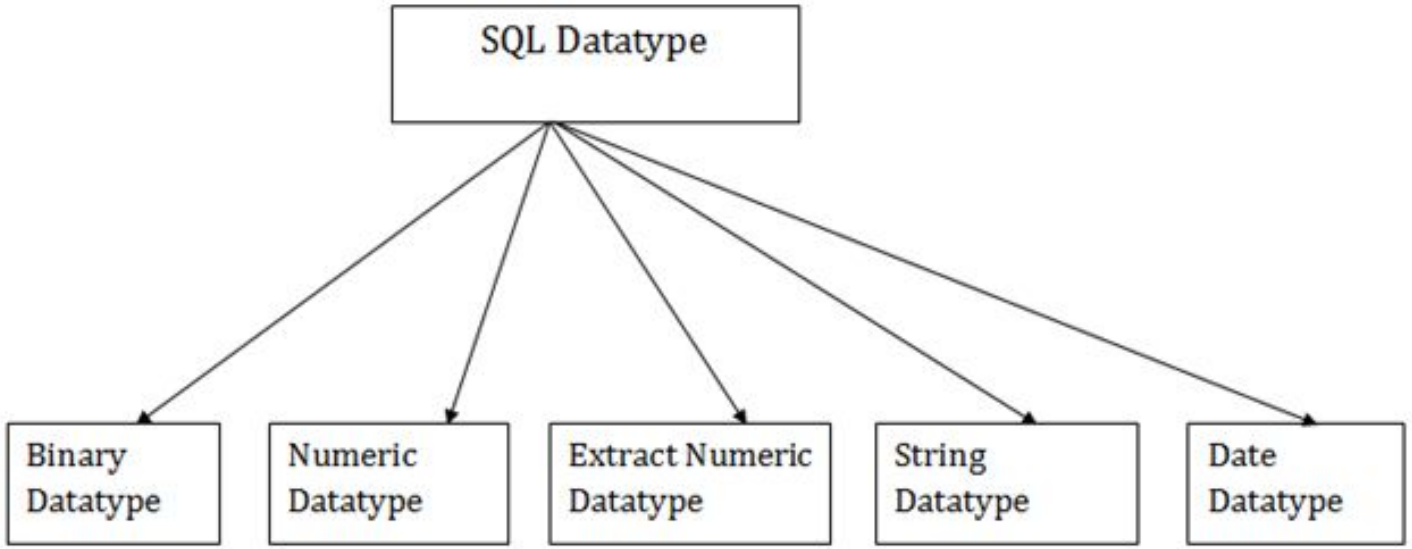
# Foreign Key Constraint

The FOREIGN KEY constraint is used to prevent actions that would destroy links between tables.
A FOREIGN KEY is a field (or collection of fields) in one table, that refers to the PRIMARY KEY in another table.
Notice that the "PersonID" column in the "Orders" table points to the "PersonID" column in the "Persons" table.
The "PersonID" column in the "Persons" table is the PRIMARY KEY in the "Persons" table.
The "PersonID" column in the "Orders" table is a FOREIGN KEY in the "Orders" table.

```
CREATE TABLE Orders (
    OrderID int NOT NULL,
    OrderNumber int NOT NULL,
    PersonID int,
    PRIMARY KEY (OrderID),
    FOREIGN KEY (PersonID) REFERENCES Persons(PersonID)
);
```

# Data types in SQL

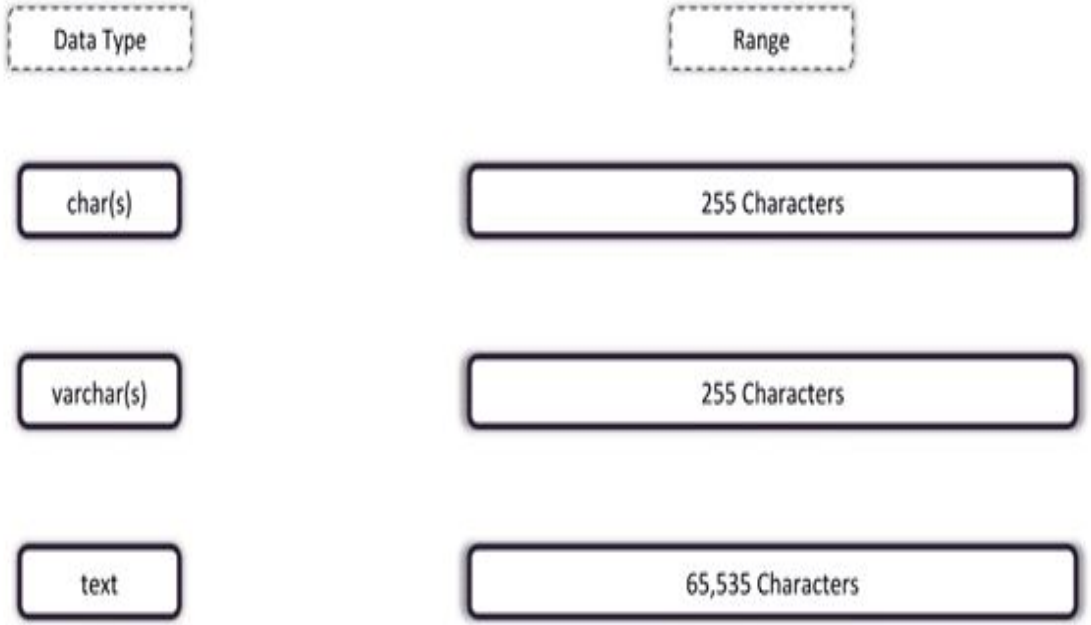
The data type of a column defines what value the column can hold: integer, character, money, date and time, binary, and so on.



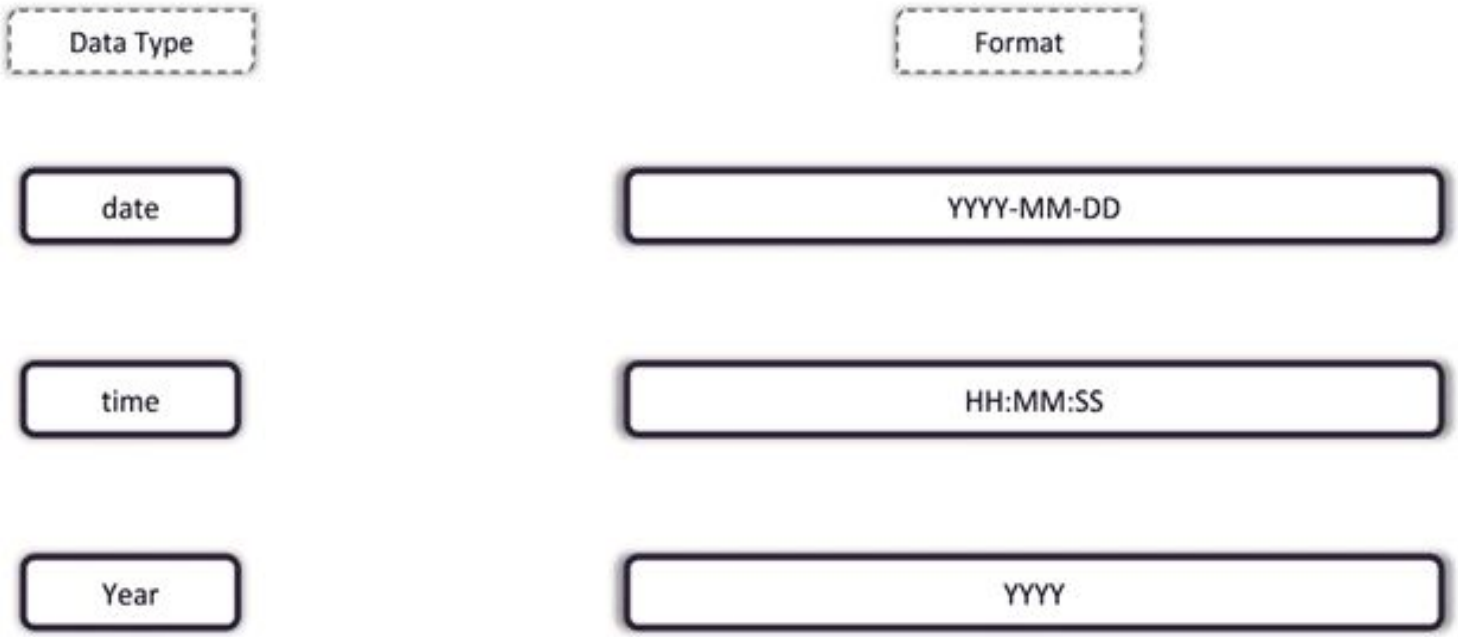
# Numerical Data Types

| Name | Storage Size | Description | Range |
|---|---|---|---|
| `integer` *or* `int` *or* `int4` | 4 bytes | typical choice | -2147483648 to +2147483647 |
| `smallint` *or* `int2` | 2 bytes | small-range | -32768 to +32767 |
| `bigint` *or* `int8` | 8 bytes | large-range | -9223372036854775808 to +9223372036854775807 |
| `serial` | 4 bytes | auto-increment | 1 to 2147483647 |
| `smallserial` | 2 bytes | small auto-increment | 1 to 32767 |
| `bigserial` | 8 bytes | large auto-increment | 1 to 9223372036854775807 |

# Character Data Types



# Date and Time Data Types



# Create Table

# CREATE

It consists of SQL commands which is used to define the database schema.

**Syntax** – *create table table_name*

**Sample Problem Statement -**

Write a query to create the Store Details table with following constraints :

1. Store Column As Primary Key
2. Store_name Column As Not Null
3. Sales Column As Check
4. Order_no Column As Unique
5. Store_location Column As Default Constraint With Default Value As Bangalore
6. City Column as Varchar
7. Pincode as int

**SQL Query**

```
Create table Store_Details(Store int primary
key,Store_Name varchar(200) not
null,Sales int
check(Sales>20),Order_No int unique,
Store_Location varchar(200) Default
"Bangalore", City varchar(200),pincode int);
```

| Store | Store_Name | Sales | Order_No | Store_Location | City | pincode |
|---|---|---|---|---|---|---|

# INSERT

# INSERT by adding values for all columns

**Syntax** - INSERT INTO table_name VALUES (val1,val2,val3,….);

Insert 10 rows into the Store Details table.

```
insert into Store_Details values(1,'Walmart',374,246,'Bentonville, Ark','Montgomery',36104),
(2,'The Kroger Co',115,240,'Cincinnati','Juneau',99801),
(3,'Costco',93,567,'Issaquah, Wash','Phoenix',85001),
(4,'The Home Depot',91,639,'Atlanta','Little Rock',72201),
(5,'Walgreens Boots Alliance',82,484,'Deerfield, Ill','Sacramento',95814),
(6,'CVS Health Corporation',79,890,'Woonsocket, R.I','Denver',80202),
(7,'Target',71,251,'Minneapolis','Hartford',06103),
(8,'Lowe Companies',63,308,'Mooresville, N.C','Dover',19901),
(9,'Albertsons Companies',59,454,'Boise, Idaho','Tallahassee',32301),
(10,'Royal Ahold Delhaize USA',43,254,'Carlisle, Pa','Atlanta',30303);
```

# INSERT by adding values for all columns

**Output -**

| | Store | Store_Name | Sales | Order_No | Store_Location | City | pincode |
|---|---|---|---|---|---|---|---|
| 1 | 1 | Walmart | 374 | 246 | Bentonville, Ark | Montgomery | 36104 |
| 2 | 2 | The Kroger Co | 115 | 240 | Cincinnati | Juneau | 99801 |
| 3 | 3 | Costco | 93 | 567 | Issaquah, Wash | Phoenix | 85001 |
| 4 | 4 | The Home Depot | 91 | 639 | Atlanta | Little Rock | 72201 |
| 5 | 5 | Walgreens Boots Alliance | 82 | 484 | Deerfield, Ill | Sacramento | 95814 |
| 6 | 6 | CVS Health Corporation | 79 | 890 | Woonsocket, R.I | Denver | 80202 |
| 7 | 7 | Target | 71 | 251 | Minneapolis | Hartford | 6103 |
| 8 | 8 | Lowe Companies | 63 | 308 | Mooresville, N.C | Dover | 19901 |
| 9 | 9 | Albertsons Companies | 59 | 454 | Boise, Idaho | Tallahassee | 32301 |
| 10 | 10 | Royal Ahold Delhaize USA | 43 | 254 | Carlisle, Pa | Atlanta | 30303 |

# Inserting by both column and values

**Syntax-** INSERT INTO table-name (column-names) VALUES (values) ;

**Code -**

```
insert into store_details(Store, Store_name, Sales, Order_No, Store_Location, City, pincode) values
(11,'Jack and Jones',525,148,'Amblipura','Bangalore',560102);
```

| | Store | Store_Name | Sales | Order_No | Store_Location | City | pincode |
|---|---|---|---|---|---|---|---|
| 1 | 1 | Walmart | 374 | 246 | Bentonville, Ark | Montgomery | 36104 |
| 2 | 2 | The Kroger Co | 115 | 240 | Cincinnati | Juneau | 99801 |
| 3 | 3 | Costco | 93 | 567 | Issaquah, Wash | Phoenix | 85001 |
| 4 | 4 | The Home Depot | 91 | 639 | Atlanta | Little Rock | 72201 |
| 5 | 5 | Walgreens Boots Alliance | 82 | 484 | Deerfield, Ill | Sacramento | 95814 |
| 6 | 6 | CVS Health Corporation | 79 | 890 | Woonsocket, R.I | Denver | 80202 |
| 7 | 7 | Target | 71 | 251 | Minneapolis | Hartford | 6103 |
| 8 | 8 | Lowe Companies | 63 | 308 | Mooresville, N.C | Dover | 19901 |
| 9 | 9 | Albertsons Companies | 59 | 454 | Boise, Idaho | Tallahassee | 32301 |
| 10 | 10 | Royal Ahold Delhaize USA | 43 | 254 | Carlisle, Pa | Atlanta | 30303 |
| 11 | 11 | Jack and Jones | 525 | 148 | Amblipura | Bangalore | 560102 |

# Inserting data in specified columns

**Syntax-** Insert into table_name( col1,col2,col3) values(v1,v2,v3);

```
Insert into store_details( Store, Store_name, Sales, City) values
(12,'H&M',676,'Mumbai');
```

| | Store | Store_Name | Sales | Order_No | Store_Location | City | pincode |
|---|---|---|---|---|---|---|---|
| 1 | 1 | Walmart | 374 | 246 | Bentonville, Ark | Montgomery | 36104 |
| 2 | 2 | The Kroger Co | 115 | 240 | Cincinnati | Juneau | 99801 |
| 3 | 3 | Costco | 93 | 567 | Issaquah, Wash | Phoenix | 85001 |
| 4 | 4 | The Home Depot | 91 | 639 | Atlanta | Little Rock | 72201 |
| 5 | 5 | Walgreens Boots Alliance | 82 | 484 | Deerfield, Ill | Sacramento | 95814 |
| 6 | 6 | CVS Health Corporation | 79 | 890 | Woonsocket, R.I | Denver | 80202 |
| 7 | 7 | Target | 71 | 251 | Minneapolis | Hartford | 6103 |
| 8 | 8 | Lowe Companies | 63 | 308 | Mooresville, N.C | Dover | 19901 |
| 9 | 9 | Albertsons Companies | 59 | 454 | Boise, Idaho | Tallahassee | 32301 |
| 10 | 10 | Royal Ahold Delhaize USA | 43 | 254 | Carlisle, Pa | Atlanta | 30303 |
| 11 | 11 | Jack and Jones | 525 | 148 | Amblipura | Bangalore | 560102 |
| 12 | 12 | H&M | 676 | NULL | Bangalore | Mumbai | NULL |

# SELECT

# SELECT an Individual column from a table

**Syntax** - select column_name from table_name

Select Store_name column from Store_details

**Code -**
select Store_Name from Store_Details

| | Store_Name |
|---|---|
| 1 | Walmart |
| 2 | The Kroger Co |
| 3 | Costco |
| 4 | The Home Depot |
| 5 | Walgreens Boots Alliance |
| 6 | CVS Health Corporation |
| 7 | Target |
| 8 | Lowe Companies |
| 9 | Albertsons Companies |
| 10 | Royal Ahold Delhaize USA |

# SELECT multiple columns from a table

**Syntax** - select column_name1, column_name2,....,column_nameN from table name

Select Store_name, Store_Location and City columns from Store_details

**Code -**

```
select Store_Name,Store_Location,City
from Store_Details
```

| | Store_Name | Store_Location | City |
|---|---|---|---|
| 1 | Walmart | Bentonville, Ark | Montgomery |
| 2 | The Kroger Co | Cincinnati | Juneau |
| 3 | Costco | Issaquah, Wash | Phoenix |
| 4 | The Home Depot | Atlanta | Little Rock |
| 5 | Walgreens Boots Alliance | Deerfield, Ill | Sacramento |
| 6 | CVS Health Corporation | Woonsocket, R.I | Denver |
| 7 | Target | Minneapolis | Hartford |
| 8 | Lowe Companies | Mooresville, N.C | Dover |
| 9 | Albertsons Companies | Boise, Idaho | Tallahassee |
| 10 | Royal Ahold Delhaize USA | Carlisle, Pa | Atlanta |

# SELECT multiple columns from a table

**Syntax** - select * from table name

Select all the columns from Store_details

**Code -** select * from Store_Details

| | Store | Store_Name | Sales | Order_No | Store_Location | City | pincode | DEPARTMENT |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | Walmart | 374 | 246 | Bentonville, Ark | Montgomery | 36104 | COSMETICS |
| 2 | 2 | The Kroger Co | 115 | 240 | Cincinnati | Juneau | 99801 | GROCERIES |
| 3 | 3 | Costco | 93 | 567 | Issaquah, Wash | Phoenix | 85001 | COSMETICS |
| 4 | 4 | The Home Depot | 91 | 639 | Atlanta | Little Rock | 72201 | GROCERIES |
| 5 | 5 | Walgreens Boots Alliance | 82 | 484 | Deerfield, Ill | Sacramento | 95814 | STATIONARY |
| 6 | 6 | CVS Health Corporation | 79 | 890 | Woonsocket, R.I | Denver | 80202 | MEDICAL |
| 7 | 7 | Target | 71 | 251 | Minneapolis | Hartford | 6103 | COSMETICS |
| 8 | 8 | Lowe Companies | 63 | 308 | Mooresville, N.C | Dover | 19901 | STATIONARY |
| 9 | 9 | Albertsons Companies | 59 | 454 | Boise, Idaho | Tallahassee | 32301 | SPORTS |
| 10 | 10 | Royal Ahold Delhaize USA | 43 | 254 | Carlisle, Pa | Atlanta | 30303 | STATIONARY |

# ALTER

# ALTER by adding column

It is used to add, modify or delete columns in an existing table

Add a column profit with datatype int in store_details

**Syntax -**

ALTER TABLE table_name
ADD column_name datatype;

alter table store_details
add profit int;

| Store | Store_Name | Sales | Order_No | Store_Location | City | pincode | profit |
|---|---|---|---|---|---|---|---|
| 1 | Walmart | 374 | 246 | Bentonville, Ark | Montgomery | 36104 | NULL |
| 2 | The Kroger Co | 115 | 240 | Cincinnati | Juneau | 99801 | NULL |
| 3 | Costco | 93 | 567 | Issaquah, Wash | Phoenix | 85001 | NULL |
| 4 | The Home Depot | 91 | 639 | Atlanta | Little Rock | 72201 | NULL |
| 5 | Walgreens Boots Alliance | 82 | 484 | Deerfield, Ill | Sacramento | 95814 | NULL |
| 6 | CVS Health Corporation | 79 | 890 | Woonsocket, R.I | Denver | 80202 | NULL |
| 7 | Target | 71 | 251 | Minneapolis | Hartford | 6103 | NULL |
| 8 | Lowe Companies | 63 | 308 | Mooresville, N.C | Dover | 19901 | NULL |
| 9 | Albertsons Companies | 59 | 454 | Boise, Idaho | Tallahassee | 32301 | NULL |
| 10 | Royal Ahold Delhaize U... | 43 | 254 | Carlisle, Pa | Atlanta | 30303 | NULL |
| 11 | Jack and Jones | 525 | 148 | Amblipura | Bangalore | 560102 | NULL |
| 12 | H&M | 676 | NULL | Bangalore | Mumbai | NULL | NULL |

# ALTER by Dropping column

Remove the column profit in table store_details

**Syntax -**
ALTER TABLE *table_name*
DROP COLUMN *column_name*;

```
alter table store_details
drop column profit
```

**NOTE:** ALTER TABLE is rarely used

| Store | Store_Name | Sales | Order_No | Store_Location | City | pincode |
|---|---|---|---|---|---|---|
| 1 | Walmart | 374 | 246 | Bentonville, Ark | Montgomery | 36104 |
| 2 | The Kroger Co | 115 | 240 | Cincinnati | Juneau | 99801 |
| 3 | Costco | 93 | 567 | Issaquah, Wash | Phoenix | 85001 |
| 4 | The Home Depot | 91 | 639 | Atlanta | Little Rock | 72201 |
| 5 | Walgreens Boots Alliance | 82 | 484 | Deerfield, Ill | Sacramento | 95814 |
| 6 | CVS Health Corporation | 79 | 890 | Woonsocket, R.I | Denver | 80202 |
| 7 | Target | 71 | 251 | Minneapolis | Hartford | 6103 |
| 8 | Lowe Companies | 63 | 308 | Mooresville, N.C | Dover | 19901 |
| 9 | Albertsons Companies | 59 | 454 | Boise, Idaho | Tallahassee | 32301 |
| 10 | Royal Ahold Delhaize USA | 43 | 254 | Carlisle, Pa | Atlanta | 30303 |
| 11 | Jack and Jones | 525 | 148 | Amblipura | Bangalore | 560102 |
| 12 | H&M | 676 | NULL | Bangalore | Mumbai | NULL |

# UPDATE

# Update using WHERE clause

**Syntax –**
**UPDATE** table_name
**SET** column1 = value1, column2 = value2, ...
**WHERE** condition;

**update store_details set store_Name="Nike" Where City="Bangalore"**

| Store | Store_Name | Sales | Order_No | Store_Location | City | pincode |
|---|---|---|---|---|---|---|
| 1 | Walmart | 374 | 246 | Bentonville, Ark | Montgomery | 36104 |
| 2 | The Kroger Co | 115 | 240 | Cincinnati | Juneau | 99801 |
| 3 | Costco | 93 | 567 | Issaquah, Wash | Phoenix | 85001 |
| 4 | The Home Depot | 91 | 639 | Atlanta | Little Rock | 72201 |
| 5 | Walgreens Boots Alliance | 82 | 484 | Deerfield, Ill | Sacramento | 95814 |
| 6 | CVS Health Corporation | 79 | 890 | Woonsocket, R.I | Denver | 80202 |
| 7 | Target | 71 | 251 | Minneapolis | Hartford | 6103 |
| 8 | Lowe Companies | 63 | 308 | Mooresville, N.C | Dover | 19901 |
| 9 | Albertsons Companies | 59 | 454 | Boise, Idaho | Tallahassee | 32301 |
| 10 | Royal Ahold Delhaize U... | 43 | 254 | Carlisle, Pa | Atlanta | 30303 |
| 11 | Nike | 525 | 148 | Amblipura | Bangalore | 560102 |
| 12 | H&M | 676 | NULL | Bangalore | Mumbai | NULL |

# Update TOP Statements

The TOP Statement to limit the number of rows that are modified in an UPDATE statement. When a TOP (*n*) clause is used with UPDATE, the update operation is performed on a random selection of '*n*' number of rows

```
UPDATE top (5) store_details set sales = 100;
```

| Store | Store_Name | Sales | Order_No | Store_Location | City | pincode |
|---|---|---|---|---|---|---|
| 1 | Walmart | 100 | 246 | Bentonville, Ark | Montgomery | 36104 |
| 2 | The Kroger Co | 100 | 240 | Cincinnati | Juneau | 99801 |
| 3 | Costco | 93 | 567 | Issaquah, Wash | Phoenix | 85001 |
| 4 | The Home Depot | 91 | 639 | Atlanta | Little Rock | 72201 |
| 5 | Walgreens Boots Alliance | 82 | 484 | Deerfield, Ill | Sacramento | 95814 |
| 6 | CVS Health Corporation | 79 | 890 | Woonsocket, R.I | Denver | 80202 |
| 7 | Target | 100 | 251 | Minneapolis | Hartford | 6103 |
| 8 | Lowe Companies | 63 | 308 | Mooresville, N.C | Dover | 19901 |
| 9 | Albertsons Companies | 59 | 454 | Boise, Idaho | Tallahassee | 32301 |
| 10 | Royal Ahold Delhaize USA | 43 | 254 | Carlisle, Pa | Atlanta | 30303 |
| 11 | Nike | 100 | 148 | Amblipura | Bangalore | 560102 |
| 12 | H&M | 100 | NULL | Bangalore | Mumbai | NULL |

# Update table with data from another table

Create another table summary with column name store having int datatype as primary key, category column as unique. Insert six records in summary table and update the category column of summary table with Order_no of store_details. (Note- Take store from 1 to 6 for summary table)

| Store | Store_Name | Sales | Order_No | Store_Location | City | pincode |
|---|---|---|---|---|---|---|
| 1 | Walmart | 100 | 246 | Bentonville, Ark | Montgomery | 36104 |
| 2 | The Kroger Co | 100 | 240 | Cincinnati | Juneau | 99801 |
| 3 | Costco | 93 | 567 | Issaquah, Wash | Phoenix | 85001 |
| 4 | The Home Depot | 91 | 639 | Atlanta | Little Rock | 72201 |
| 5 | Walgreens Boots Alliance | 82 | 484 | Deerfield, Ill | Sacramento | 95814 |
| 6 | CVS Health Corporation | 79 | 890 | Woonsocket, R.I | Denver | 80202 |
| 7 | Target | 100 | 251 | Minneapolis | Hartford | 6103 |
| 8 | Lowe Companies | 63 | 308 | Mooresville, N.C | Dover | 19901 |
| 9 | Albertsons Companies | 59 | 454 | Boise, Idaho | Tallahassee | 32301 |
| 10 | Royal Ahold Delhaize USA | 43 | 254 | Carlisle, Pa | Atlanta | 30303 |
| 11 | Nike | 100 | 148 | Amblipura | Bangalore | 560102 |
| 12 | H&M | 100 | NULL | Bangalore | Mumbai | NULL |

| Store | category |
|---|---|
| 3 | 99 |
| 1 | 100 |
| 5 | 103 |
| 2 | 105 |
| 4 | 108 |
| 6 | 110 |

# Update table with data from another table

```
UPDATE summary SET category = (SELECT order_No FROM store_details
WHERE store_details.store = summary.store)
WHERE EXISTS (SELECT order_no FROM store_details WHERE store_details.store = summary.store);
```

OUTPUT

| Store | category |
|---|---|
| 2 | 240 |
| 1 | 246 |
| 5 | 484 |
| 3 | 567 |
| 4 | 639 |
| 6 | 890 |

# Update top 10 records of table

CODE

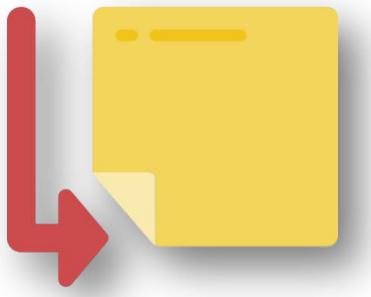
```
alter table sales
add Profit Varchar(5),Loss
varchar(5)

set rowcount 10
update sales
set Profit='YES',Loss='NO'
where Weekly_Sales>30000
SET ROWCOUNT 0;
select * from sales
```

OUTPUT

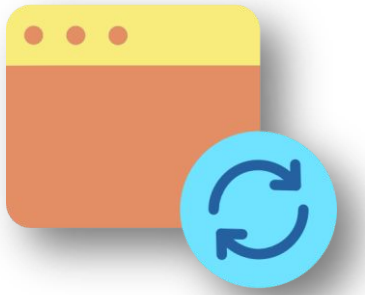
| | Store | Dept | Date | Weekly_Sales | IsHoliday | Profit | Loss |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 2010-05-02 | 24924.50 | FALSE | NULL | NULL |
| 2 | 1 | 1 | 2010-12-02 | 46039.49 | TRUE | YES | NO |
| 3 | 1 | 1 | 2010-02-19 | 41595.55 | FALSE | YES | NO |
| 4 | 1 | 1 | 2010-02-26 | 19403.54 | FALSE | NULL | NULL |
| 5 | 1 | 1 | 2010-05-03 | 21827.90 | FALSE | NULL | NULL |
| 6 | 1 | 1 | 2010-12-03 | 21043.39 | FALSE | NULL | NULL |
| 7 | 1 | 1 | 2010-03-19 | 22136.64 | FALSE | NULL | NULL |
| 8 | 1 | 1 | 2010-03-26 | 26229.21 | FALSE | NULL | NULL |
| 9 | 1 | 1 | 2010-02-04 | 57258.43 | FALSE | YES | NO |
| 10 | 1 | 1 | 2010-09-04 | 42960.91 | FALSE | YES | NO |
| 11 | 1 | 1 | 2010-04-16 | 17596.96 | FALSE | NULL | NULL |
| 12 | 1 | 1 | 2010-04-23 | 16145.35 | FALSE | NULL | NULL |
| 13 | 1 | 1 | 2010-04-30 | 16555.11 | FALSE | NULL | NULL |
| 14 | 1 | 1 | 2010-07-05 | 17413.94 | FALSE | NULL | NULL |
| 15 | 1 | 1 | 2010-05-14 | 18926.74 | FALSE | NULL | NULL |

# MERGE

MERGE is the combination of INSERT, DELETE, and UPDATE statements.



Insert



Update

Delete

# MERGE

It is a combination of insert, delete and update statements. If there is a Source table and a Target table that are to be merged, then with the help of MERGE statement, all the three operations can be performed at once.

Create two tables named source table and target table with columns Productid, Product name and Price . Now insert the values into the same.

| | ProductID | ProductName | Price |
|---|---|---|---|
| 1 | 1 | Table | 100.00 |
| 2 | 2 | Desk | 80.00 |
| 3 | 3 | Chair | 50.00 |
| 4 | 4 | Computer | 300.00 |

Source Table

| ProductID | ProductName | Price |
|---|---|---|
| 1 | Table | 100.00 |
| 2 | Desk | 180.00 |
| 5 | Bed | 50.00 |
| 6 | Cupboard | 300.00 |

Target Table

# MERGE

Insert data using merge

On following tables Insert data using merge

```
MERGE TargetProducts AS Target
USING SourceTableAS Source
ON Source.ProductID = Target.ProductID
WHEN NOT MATCHED BY Target THEN
INSERT (ProductID,ProductName, Price)
VALUES (Source.ProductID,Source.ProductName, Source.Price);
```

| ProductID | ProductName | Price |
|---|---|---|
| 1 | Table | 100.00 |
| 2 | Desk | 180.00 |
| 5 | Bed | 50.00 |
| 6 | Cupboard | 300.00 |
| 3 | Chair | 50.00 |
| 4 | Computer | 300.00 |

# MERGE

Update

On following tables Update data using merge

```
MERGE TargetProducts AS Target
  USING SourceTableAS Source
  ON Source.ProductID = Target.ProductID
  WHEN MATCHED THEN UPDATE SET
    Target.ProductName= Source.ProductName ,
    Target.Price= Source.Price;
```

| ProductID | ProductName | Price |
|---|---|---|
| 1 | Table | 100.00 |
| 2 | Desk | 80.00 |
| 5 | Bed | 50.00 |
| 6 | Cupboard | 300.00 |
| 3 | Chair | 50.00 |
| 4 | Computer | 300.00 |

# MERGE

Delete

On following tables delete data using merge

```
MERGE TargetProducts AS Target
USING SourceTableAS Source
ON Source.ProductID = Target.ProductID
WHEN NOT MATCHED BY Source THEN
DELETE;
```

| ProductID | ProductName | Price |
|---|---|---|
| 1 | Table | 100.00 |
| 2 | Desk | 80.00 |
| 3 | Chair | 50.00 |
| 4 | Computer | 300.00 |

# DELETE

It is used to delete existing records in a table

**Syntax** - DELETE FROM table_name WHERE condition;

Delete the records from store_details where sales = 100

```
DELETE FROM store_details WHERE Sales=100;
```

| Store | Store_Name | Sales | Order_No | Store_Location | City | pincode |
|---|---|---|---|---|---|---|
| 3 | Costco | 93 | 567 | Issaquah, Wash | Phoenix | 85001 |
| 4 | The Home Depot | 91 | 639 | Atlanta | Little Rock | 72201 |
| 5 | Walgreens Boots Alliance | 82 | 484 | Deerfield, Ill | Sacramento | 95814 |
| 6 | CVS Health Corporation | 79 | 890 | Woonsocket, R.I | Denver | 80202 |
| 8 | Lowe Companies | 63 | 308 | Mooresville, N.C | Dover | 19901 |
| 9 | Albertsons Companies | 59 | 454 | Boise, Idaho | Tallahassee | 32301 |
| 10 | Royal Ahold Delhaize USA | 43 | 254 | Carlisle, Pa | Atlanta | 30303 |

# TRUNCATE

It is used to delete an existing data in a table , except the table itself.

**Syntax** - TRUNCATE TABLE table_name;

Drop the existing data from the store_details

Truncate table store_details

| Store | Store_Name | Sales | Order_No | Store_Location | City | pincode |
|---|---|---|---|---|---|---|

# DROP

It is used to drop an existing table in a database

**Syntax** - DROP TABLE table_name;

Drop the existing table store_details

```
drop table store_details
```

```
Invalid object name 'store_details'.
```