

# 10.1 Simple and complex types



This section has been set as optional by your instructor.

## Simple types

Since the 1980s, relational database products have supported six broad categories of data types:

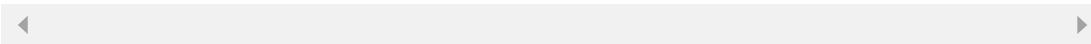
- **Integer** types represent positive and negative integers.
- **Decimal** types represent numbers with fractional values.
- **Character** types represent textual characters. Character types may be either fixed-length or variable-length strings, consisting of either single-byte (ASCII) or double-byte (Unicode) characters.
- **Time** types represent date, time, or both. Some time types include a time zone or specify a time interval.
- **Binary** types store data exactly as the data appears in memory or computer files, bit for bit. Ex: Binary types may be used to store images.
- **Semantic** types are based on other types but have a special meaning and functions. Ex: MONEY has decimal values representing currency. BOOLEAN has values zero and one representing false and true. UUID has string values representing Universally Unique Identifiers, such as `a0eebc99-9c0b-4ef8-bb6d-6bb9bd380a11`. ENUM has a fixed set of string values specified by the database designer, such as 'red', 'green', 'blue'.

The above types have relatively simple internal structures and thus are called **simple data types**. Ex: A character value consists of a series of individual characters. Ex: A date value has three parts, year, month, and day.

Database functions can decompose the internal structure into separate values. Ex: If BirthDate has type date, the SQL function `month(BirthDate)` might return the birth month. From the perspective of arithmetic and comparison operators, however, the internal structure is ignored and each value is considered atomic.

Table 10.1.1: Simple type examples implemented by various databases.

	MySQL	Oracle Database	PostgreSQL
Integer	BIT TINYINT SMALLINT MEDIUMINT BIGINT	INT NUMBER	BIT SMALLINT INTEGER BIGINT
Decimal	FLOAT DOUBLE DECIMAL	FLOAT NUMBER	REAL NUMERIC DECIMAL
Character	CHAR VARCHAR TEXT	CHAR VARCHAR2 LONG	CHAR VARCHAR TEXT
Time	DATE DATETIME TIMESTAMP	DATE TIMESTAMP TIMESTAMP WITH TIMEZONE INTERVAL	DATE TIME TIMESTAMP INTERVAL
Binary	TINYBLOB MEDIUMBLOB LONGBLOB	BLOB BFILE RAW	BYTEA
Semantic	ENUM	UROWID	MONEY BOOLEAN UUID


[Feedback?](#)
**PARTICIPATION ACTIVITY**

10.1.1: Simple types.



- 1) NUMERIC is a decimal type supported by Oracle Database.

True  
 False

**Correct**

NUMERIC is supported by PostgreSQL, not Oracle Database. The Oracle Database NUMBER type is equivalent to the PostgreSQL NUMERIC type.



- 2) 3.1415 can be stored as an INT type in Oracle

**Correct**


- Database.
- True  
 False
- 3) 1/29/2020 14:30:00 might represent a DATETIME value in MySQL.
- True  
 False
- 4) Photographs are stored with the BYTEA type in PostgreSQL.
- True  
 False

**Correct**

MySQL supports the DATETIME type. Since 1/29/2020 represents a date and 14:30:00 represents a time, 1/29/2020 14:30:00 might represent a DATETIME value.



**Correct**

Photographs are usually stored as a binary type. BYTEA is the only binary type in PostgreSQL.



[Feedback?](#)

## Complex types

As relational database adoption increased in the 1980s, the need for additional types became apparent. Ex: A spatial type might represent a single point as an X and Y value. Ex: A composite type representing a full name might contain three simple types representing first, middle, and last names. Newer types like spatial and composite have a rich internal structure and are called **complex data types**.

Most complex types fall into one of four categories:

- **Collection** types include several distinct values of the same base type, organized as a set or an array.
- **Document** types contain textual data in a structured format such as XML or JSON.
- **Spatial** types store geometric information, such as lines, polygons, and map coordinates.
- **Object** types support object-oriented programming constructs, such as composite types, methods, and subtypes.

Research on complex types began in 1986 with the POSTGRES project at the University of California, Berkeley. POSTGRES was released as PostgreSQL in 1997 and now supports complex types in all four categories. In the 1990s, commercial products such as Oracle Database and SQL Server also added support for complex types.

From the perspective of the database system, complex types, like simple types, are atomic and stored as one value per cell.

Table 10.1.2: Complex type examples implemented by various databases.

	MySQL	Oracle Database	PostgreSQL
Collection	SET	CREATE TYPE TypeName AS VARRAY(n) OF basetype	basetype[n] basetype ARRAY[N]
Document	JSON	XMLTYPE	XML JSON
Spatial	POINT MULTIPOINT POLYGON MULTIPOLYGON	SDO_GEOGRAPHY SDO_GEOASTER	POINT LINE POLYGON CIRCLE
Object	none	CREATE TYPE TypeName AS OBJECT ... CREATE TYPE TypeName AS BODY ...	CREATE TYPE TypeName AS ...



[Feedback?](#)

**PARTICIPATION ACTIVITY**

10.1.2: Complex types.



- 1) Which database might have an unordered set of values { 'apple', 'orange', 'banana' } in a cell of a table?

**Correct**



In MySQL, a value of type SET consists of any subset of a fixed set of elements.

- MySQL
- Oracle Database
- PostgreSQL

- 2) What kind of type is the



following value?

```
<menu>
  <selection>
    <name>Greek
    salad</name>

    <price>$13.90</price>
      <text>Cucumbers,
    tomatoes, onions, and
    feta cheese</text>
  </selection>
  <selection>
    <name>Turkey
    sandwich</name>

    <price>$9.00</price>
      <text>Turkey,
    lettuce, tomato on
    choice of bread</text>
  </selection>
</menu>
```

- Collection
  - Document
  - Object
- 3) Which database has both XML and JSON types?
- MySQL
  - Oracle Database
  - PostgreSQL
- 4) Which database does not support object types?
- MySQL
  - Oracle Database
  - PostgreSQL

### Correct

The value is a string of characters in XML format. XML is a common document type.



### Correct

PostgreSQL supports both XML and JSON types.



### Correct

MySQL does not support any object-oriented types.

[Feedback?](#)

## User-defined types

A **system-defined type**, also known as a **built-in type**, is provided by the database as a reserved keyword. Ex: FLOAT, CHAR, ENUM, SET, JSON, and POINT are MySQL system-defined types.

A **user-defined type** is created by a database designer or administrator with the CREATE TYPE statement. The **CREATE TYPE** statement specifies the type name and a **base type** that defines

the implementation. The base type can be either system-defined or user-defined. Ex: `CREATE TYPE Meters AS REAL` creates a new type named Meters with base type REAL. Although Meters is implemented as REAL, the two types are different and cannot be directly compared.

User-defined data types appear after column names in CREATE TABLE statements, just like system-defined types. User-defined types can be either simple or complex. Ex: CREATE TYPE is used to create simple enumerated types in PostgreSQL and complex array types in Oracle Database.

CREATE TYPE is specified in the SQL standard and supported in Oracle Database, PostgreSQL, SQL Server, and DB2. However, syntax and capabilities vary. MySQL does not support the CREATE TYPE statement.

**PARTICIPATION ACTIVITY**

## 10.1.3: PostgreSQL CREATE TYPE statement.



1 2 3 ← ✓ 2x speed

```
CREATE TYPE Size AS ENUM('Small', 'Medium', 'Large', 'X-Large');

CREATE TABLE Clothing (
    ItemNumber INTEGER,
    ItemName VARCHAR(20),
    ItemSize Size
);

INSERT INTO Clothing
VALUES (3281, 'Sahara sun bonnet', 'Small')
       (3400, 'Zephyr shorts', 'XLarge');
```

Clothing

ItemNumber	ItemName	ItemSize
3281	Sahara sun bonnet	Small
3400	Zephyr shorts	XLarge

The INSERT statement inserts rows with Size values 'Small' and 'XLarge'.

Captions ^

1. The CREATE TYPE statement creates a user-defined type Size, based on the system-defined type ENUM.
2. The Size type can be used in a CREATE TABLE statement.
3. The INSERT statement inserts rows with Size values 'Small' and 'XLarge'.

**Feedback?**

**PARTICIPATION ACTIVITY**

## 10.1.4: User-defined types.



1) Which database does not support the CREATE TYPE statement?

- MySQL
- Oracle Database
- PostgreSQL

2) User-defined types can be:

- Simple types only
- Complex types only
- Either simple or complex types.

3) System-defined types can be:

- Simple types only
- Complex types only
- Either simple or complex types



### Correct

MySQL does not support the CREATE TYPE statement and does not allow users to create and name types.



### Correct

Depending on the database system, the CREATE TYPE statement can create either simple or complex types.



### Correct

Both simple and complex types can be system-defined. Ex: INTEGER and DECIMAL are system-defined simple types. JSON and XML are system-defined complex types.

[Feedback?](#)

Exploring further:

- [MySQL types](#)
- [Oracle Database types](#)
- [PostgreSQL types](#)

How

was this section?



[Provide feedback](#)

