10.2 Collection types

This section has been set as optional by your instructor.

Collection types

A collection type is defined in terms of a base type. Each collection type value contains zero, one, or many base type values. Ex: `QuarterlySales INTEGER ARRAY[4]` defines a column QuarterlySales with the collection type ARRAY and base type INTEGER. Base type values are called ***elements***.

Collections include four complex types:

- Elements of a ***set*** value cannot be repeated and are not ordered.
- Elements of a ***multiset*** value can be repeated and are not ordered.
- Elements of a ***list*** value can be repeated and are ordered.
- An ***array*** is an indexed list. Each element of an array value can be accessed with a numeric index.

The SQL standard includes multiset and array types but not set and list types. Most databases support only one or two collection types, and implementations vary greatly. Ex: In the MySQL set type, the base type must be a fixed group of character strings. In Informix, the base type can be any type, including complex types. A set value can be NULL in MySQL but not in Informix.

Table 10.2.1: Collection type support.

	Set	Mulitset	List	Array
MySQL	✓	-	-	-
Oracle Database	-	-	-	✓
PostgreSQL	-	-	-	✓
SQL Server	-	-	-	-
DB2	-	-	-	-
Informix	✓	✓	✓	-

Feedback?

PARTICIPATION ACTIVITY 10.2.1: Collection types.

1) The only difference between the multiset and list types is that elements are ordered in a list value and not in a multiset value.

- (●) True
- ( ) False

Correct

Elements can be repeated in both multiset and list values. However, elements are ordered in a list value and not in a multiset value.

2) In the Informix set type, the base type must be a group of character strings.

- ( ) True
- (●) False

Correct

Informix set, multiset, and list types can be based on any type.

3) In all implementations, the base type of a collection can be any supported type.

- ( ) True
- (●) False

Correct

Collection type implementations vary across databases. In the MySQL set type, the base type must be a group of character strings.

4) In some implementations, the base type of a collection can be a complex type.

- (●) True
- ( ) False

Correct

Most databases allow complex base types. Informix collections can have any base type. The base type for a PostgreSQL array can be some, but not all, complex types.

5) Collection type implementations always conform to the SQL standard.

- ( )

Correct

Not all collection types are standardized in SQL. Even when a collection type is standardized, such as multiset and array types, implementations vary across databases.

True
False

Feedback?

Set type

The MySQL SET type is similar to the ENUM type — both have a base type consisting of character strings. Each ENUM value must contain exactly one element, while each SET value may contain zero, one, or many elements.

A SET type is defined with the SET keyword followed by the base type strings. Ex: `SET('apple', 'banana', orange')` is the type consisting of elements 'apple', 'banana', and 'orange'. A SET value is specified as a string with commas between each element and no blank spaces. Ex: `'apple,orange'` is a value of `SET('apple', 'banana', orange')`.

A SET value can contain no elements. A value with no elements represents the empty set and is not the same as a NULL value.

Internally, each SET value is represented as a series of bits. Each bit corresponds to a specific element. When a bit is one, the corresponding element is included in a value. When a bit is zero, the corresponding element is not included. A base type can have at most 64 elements, so each SET value requires at most 64 bits, or eight bytes.

PARTICIPATION ACTIVITY 10.2.2: MySQL set type.

1 2 3 2x speed

```
CREATE TABLE Employee (
   ID INTEGER,
   Name VARCHAR(20),
   Language SET('English', 'French', 'Spanish', 'Mandarin', 'Japanese'),
   PRIMARY KEY (ID)
);
INSERT INTO Employee (ID, Name, Language)
VALUES (2538, 'Lisa Ellison', 'English,Spanish'),
       (6381, 'Maria Rodriguez', 'English,Spanish,Japanese'),
       (7920, 'Jiho Chen', 'Mandarin');
```

Employee

ID	Name	Language
2538	Lisa Ellison	English, Spanish
6381	Maria Rodriguez	English, Spanish, Japanese
7920	Jiho Chen	Mandarin

```
SELECT *
FROM Employee
WHERE Language LIKE '%Spanish%';
```

Result

ID	Name	Language

2538	Lisa Ellison	English, Spanish
6381	Maria Rodriguez	English, Spanish, Japanese

SET values can be compared with string functions.

Captions

1. A SET is defined with the SET keyword followed by a list of elements.
2. SET values are specified as strings containing elements separated by commas. The string cannot contain blanks.
3. SET values can be compared with string functions.

Feedback?

PARTICIPATION ACTIVITY 10.2.3: Set type.

Refer to the MySQL table created by this statement:

```
CREATE TABLE Part (
  PartNumber INTEGER,
  Material SET ('Steel', 'Copper', 'Aluminum', 'Zinc'),
  PRIMARY KEY (PartNumber)
);
```

1) Which string is a correct value of the Material column?

- 'Steel, Copper, Zinc'
- 'Steel,Copper,Zinc'
- ('Steel', 'Copper', 'Zinc')

Correct

Each SET value is a single character string enclosed in quotes with no spaces.

2) Which query selects all parts containing copper?

-
```
SELECT
PartNumber
FROM Part
WHERE Material =
'Copper';
```
-
```
SELECT
PartNumber
FROM Part
WHERE Material =
'%Copper%';
```

Correct

When compared to another string with the LIKE operator, '%' is interpreted as a wildcard. Since 'Copper' may appear in the middle of a Material string, '%' must appear both before and after 'Copper'.

```
SELECT
PartNumber
FROM Part
WHERE Material
LIKE '%Copper%';
```

3) If a SET type has 25 elements, how many bytes does each set value require?

- Two bytes
- Four bytes
- Eight bytes

Correct

Four bytes contain 32 bits and therefore can represent a set with 25 elements. Three bytes contains 24 bits and is too small to represent 25 elements.

Feedback?

Array type

PostgreSQL specifies array types by appending a pair of brackets to any base type. A number in the brackets indicates the array size. Ex: `INTEGER[4]` is an array type consisting of four integers. Optionally, the keyword ARRAY can appear between the base type and brackets. Ex: `INTEGER ARRAY[4]` is equivalent to `INTEGER[4]`. If no number appears in the brackets, array size is variable, up to a system maximum.

An array value is specified as a string with comma-separated values within braces. Ex: `'{2, 5, 11, 6}'` is a value of `INTEGER[4]`. An individual array element is specified with an index within brackets. Ex: `WHERE MonthlyHours[2] > 100` selects all rows in which the second element of the MonthlyHours column exceeds 100.

A multidimensional array type can be specified with multiple bracket pairs. Ex: `INTEGER[4][9]` is an integer array with four rows and nine columns. A multidimensional array value is specified with nested braces. Ex: `'{ {2, 5}, {11, 6}, {45, 0} }'` is a value of `INTEGER[3][2]`.

PARTICIPATION ACTIVITY 10.2.4: PostgreSQL array type.

1 2 3 2x speed

```
CREATE TABLE Employee (
  ID INTEGER,
  Name VARCHAR(20),
  QuarterlySales INTEGER[4],
  PRIMARY KEY (ID)
);

INSERT INTO Employee (ID, Name, QuarterlySales)
VALUES (2538, 'Lisa Ellison', '{ 1450, 2020, 900, 5370 }'),
       (6381, 'Maria Rodriguez', '{ 3340, 800, 1700, 6400 }'),
       (7920, 'Jiho Chen', '{ 0, 3900, 8000, 320 }');
```

Employee

ID	Name	QuarterlySales
2538	Lisa Ellison	{ 1450, 2020, 900, 5370 }
6381	Maria Rodriguez	{ 3340, 800, 1700, 6400 }
7920	Jiho Chen	{ 0, 3900, 8000, 320 }

```
SELECT *
FROM Employee
WHERE QuarterlySales[2] > 1000;
```

Result

ID	Name	QuarterlySales
2538	Lisa Ellison	{ 1450, 2020, 900, 5370 }
7920	Jiho Chen	{ 0, 3900, 8000, 320 }

QuarterlySales[2] refers to the second element of each value in the QuarterlySales column.

Captions

1. The QuarterlySales column is an array of four integers, representing employee sales in each of four quarters.
2. Array values are specified as elements separated by commas and within braces.
3. QuarterlySales[2] refers to the second element of each value in the QuarterlySales column.

Feedback?

In Oracle Database, an array is a user-defined type, created with the CREATE TYPE statement. Ex: The statement `CREATE TYPE Numbers VARRAY(4) OF INTEGER` creates a four-integer array type called Numbers.

PARTICIPATION ACTIVITY 10.2.5: Array type.

Match the PostgreSQL fragment with the description.

DATE [100]	An array type containing 100 elements The number inside the brackets indicates the array size.	Correct

TEXT [10] [100]	A two-dimensional array type Two pairs of brackets indicates a two-dimensional array.	Correct
DATE [ ]	An array type containing an unspecified number of elements When no number appears inside the brackets, the number of elements in a value is limited by a system default.	Correct
{ 'Copper', 'Iron', 'Copper', 'Zinc', 'Zinc' }	A one-dimensional array value A one-dimensional array value consists of elements separated by commas and enclosed by a single pair of braces.	Correct
{ { 'Copper', 'Iron' }, { 'Copper', 'Zinc' }, { 'Zinc', 'Aluminum' } }	A two-dimensional array value A two-dimensional array value contains nested braces.	Correct

Reset

Feedback?

CHALLENGE ACTIVITY 10.2.1: Collection types.

379958.2369558.qx3zqy7

Jump to level 1

1 2 3 4 5

Select all valid array values of type `TEXT[3][2]`.
Ex: '{ {2, 3, 4, 5}, {11, 12, 13, 14} }' is a value of `INTEGER[2][4]`.

(1) ☐ '{ indigo, violet, gray, magenta, teal, ochre }'
(2) ☐ '{ {ochre, gray, beige}, {indigo, yellow, sienna} }'
(3) ☑ '{ {ochre, gold}, {tan, brick}, {red, green} }'
(4) ☑ '{ {ochre, sienna}, {brick, red}, {beige, gray} }'
(5) ☐ { {yellow, orange}, {red, tan}, {indigo, beige} }
(6) ☐ '{ {orange, gold, magenta}, {brown, beige, sienna} }'

1 2 3 4 **5**

Check Next **Done**. Click any level to practice more. Completion is preserved.

✔ Expected: (3) and (4)

(3) and (4) are text arrays with 2 columns and 3 rows, so are values of `TEXT[3][2]`.

Invalid values:
(1) does not have nested braces, so is a one-dimensional array.
(2) and (6) are text arrays with 3 columns and 2 rows.
(5) is not a string with comma-separated values within nested braces.

Feedback?

Exploring further:

- MySQL set type
- PostgreSQL array type

How was this section? **Provide feedback**