Text data types

DATA TYPES AND FUNCTIONS IN SNOWFLAKE

Jake Roach
Field Data Engineer

Data types and functions in Snowflake

Snowflake is used by more than 10,000 organizations globally

Data Types

Manipulating Data

Advanced Techniques

[1] https://www.snowflake.com/en/company/overview/about-snowflake/

Snowflake text data

Text data is one of the most popular data types in Snowflake

- Categorical values
- Messages
- Phone numbers
- Unstructured text

```
-- Text data might be categorical
'Software Engineer'

-- Messages are also text data
'Hello, World!'

-- As well as phone numbers
'931-663-0164'

-- And even things like this!
'Taylormade,Titleist,Ping'
```

Text data types

```
CREATE TABLE TABLE_NAME (
    message VARCHAR(100),
    phone_number TEXT,
    brands STRING
);
```

Define column with `VARCHAR(...)`

- `TEXT`
- `STRING`

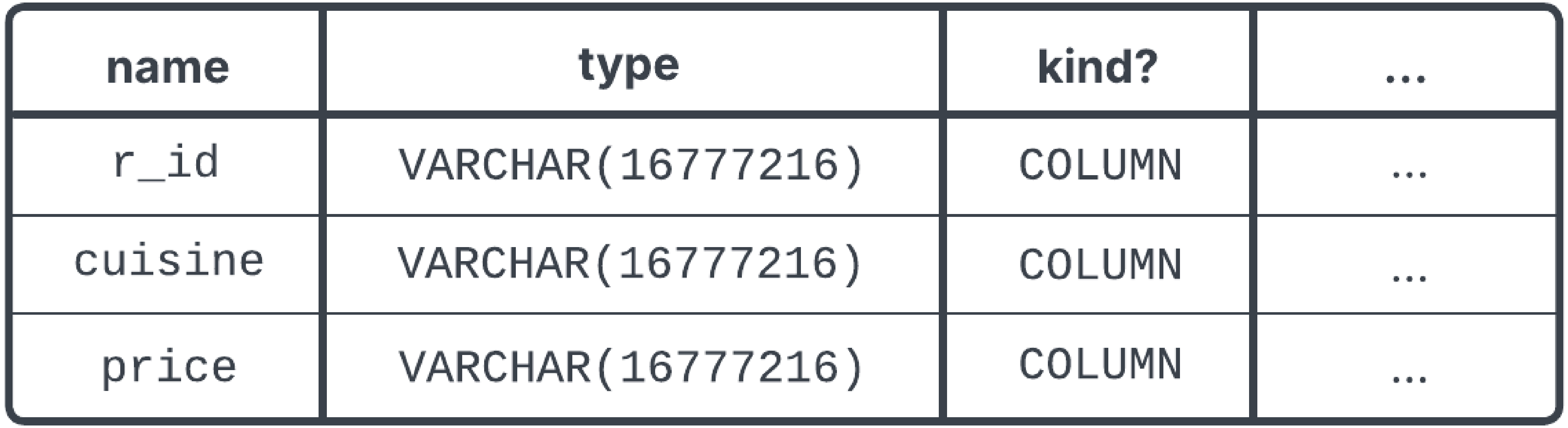
```
    message      |  phone_number  |              brands
---------------- | -------------- | ----------------------------
 Hello, World!   |  931-663-0164  |  Taylormade, Titleist, Ping
 Goodnight!      |  617-256-3086  |  Apple, Windows, Linux
```

Columns that store text data can look quite different!

Describing a Snowflake table

```
DESCRIBE TABLE <SCHEMA>.<table_name>;
```

- Outputs information about each column in the table
- `name` , `type` , `default` , ...



name	type	kind?	...
r_id	VARCHAR(16777216)	COLUMN	...
cuisine	VARCHAR(16777216)	COLUMN	...
price	VARCHAR(16777216)	COLUMN	...

Text data in queries

```
SELECT
    <field>,
    <another-field>,
    -- Can be used in a SELECT
    '<value>'
FROM SCHEMA.TABLE_NAME;
```

```
SELECT
    *
FROM SCHEMA.TABLE_NAME
-- Can be used in a WHERE clause
WHERE FIELD = '<value>';
```

- Text values can be used in different parts of a SQL query
- `SELECT`
- `WHERE`
- `CASE`

`'...'` Single Quotes

- Case-sensitive

Manipulating text data

Function	Description
`LENGTH`	Find the number of characters in a text value.
`TRIM`	Remove a sequence of characters at the beginning or end of a string.
`RTRIM`	Same functionality as `TRIM` , only applied to right-side of the string.
`LTRIM`	Same functionality as `TRIM` , only applied to left-side of the string.
`SPLIT`	Split a string using some delimiter, like a comma.
`SEARCH`	Search a string for a certain value.
`CAST` , `::`	Convert other data types into a `VARCHAR` data type.

Let's practice!

Numeric data types

DATA TYPES AND FUNCTIONS IN SNOWFLAKE

Jake Roach
Field Data Engineer

Numeric data types

`NUMBER`

- Control the size of numeric values
- `DECIMAL` , `INTEGER` are synonymous with `NUMBER`

```
126            -- Integers
7.99           -- Dollar amounts
.000191        -- Decimals
-46.88         -- Negative values
563719100      -- ID's


...


1.234E+2       -- Numeric constants
```

Precision and scale

`NUMBER(<precision>, <scale>)`

Precision is the **total number** of digits in the number

Scale is the number of digits **after** the decimal point

```
-- Precision: 4, Scale: 2
NUMBER(4, 2)


1.75    -- Good!
624.99  -- Won't fit
```

Precision and scale

```
-- Precision: 3, Scale: 0
NUMBER(3, 0)


42     -- Good!
1000   -- Won't fit
41.99  -- Won't fit
```

- Default `NUMBER` is `NUMBER(38, 0)`

Defining numeric data types

```
CREATE TABLE sales (
    id NUMBER,                  -- Traditional numeric data
    price NUMBER(5, 2),         -- Dollar values up to $999.99
    quantity NUMBER(4, 0)       -- Integer values less than 10,000
);
```

```
DESCRIBE TABLE schema.table;
```

```
   name    |     type     |   kind   | ...
---------- | ------------ | -------- | ---
    id     | NUMBER(38,0) |  COLUMN  | ...
   price   | NUMBER(5,2)  |  COLUMN  | ...
 quantity  | NUMBER(4,0)  |  COLUMN  | ...
```

FLOAT data type

- Used in Scientific and statistical calculations
- "Approximate" numeric type
- Possibility of rounding errors
- **Fast and can handle extreme scale!**

`0.0000000120056`

```sql
CREATE TABLE metrics (
    id NUMBER,
    accuracy FLOAT
);
```

```sql
DESCRIBE TABLE schema.table;
```

```
   name     |      type      | ...
----------- | -------------- | ---
    id      |  NUMBER(38,0)  | ...
 accuracy   |  FLOAT         | ...
```

Manipulating numeric data

- Numeric data can be manipulated in a number of different ways

- **Comparing values** using tools such as `=` , `!=` , `<` , or `>` .
- Performing **arithmetic operations,** such as `+` , `-` , `/` , or `*` .
- **Aggregating data** with `SUM` , `AVG` , or `COUNT` .
- Calculating **summary statistics** with `MEDIAN` , `STDDEV` , and `CORR` .

- *... and tons more!*

Let's practice!

DATA TYPES AND FUNCTIONS IN SNOWFLAKE

Datetime data types

DATA TYPES AND FUNCTIONS IN SNOWFLAKE

Jake Roach
Field Data Engineer

DATE data types

- Date values **store the day on which something occurred**

```
2023-04-01
YYYY-MM-DD
```

```sql
CREATE TABLE sales (
    transaction_date DATE  -- Define a column of type `DATE`
);
```

DATE data types

```sql
SELECT
    TO_DATE('2023-04-01') AS the_date, -- Convert a string into a DATE
    '2023-04-01'::DATE AS casted_date  -- Casted using <value>::DATE
;
```

```
  the_date  | casted_date
----------- | ------------
 2023-04-01 | 2023-04-01
```

Time data types

- Captures the **hours, minutes, and seconds that some event occurred**

```
08:24:08
HH:MM:SS
```

```
CREATE TABLE sales (
    transaction_time TIME  -- Create a column with the `TIME` keyword
);
```

- Define a `TIME` column

Time data types

```
SELECT
    TO_TIME('08:24:04') AS the_time,
    '08:24:04'::TIME AS casted_time
;
```

```
|   the_time   | casted_time |
| ----------- | ------------ |
|   08:24:08   |   08:24:08   |
```

- `TO_TIME` function
- Cast using `::`

Timestamp data types

- `TIMESTAMP` captures both date and time

```
2023-04-01 08:24:04
YYYY-MM-DD HH:MM:SS
```

```sql
CREATE TABLE (
    transaction_timestamp TIMESTAMP
);
```

Timestamp data types

```
SELECT
    TO_TIMESTAMP('2023-04-01 08:24:04') AS the_timestamp,
    '2023-04-01 08:24:04'::TIMESTAMP AS casted_timestamp
;
```

- Convert a string to a timestamp using `TO_TIMESTAMP` or `<value>::TIMESTAMP`
- Most common way to store datetime data

```
       the_timestamp      |     casted_timestamp
--------------------------|--------------------------
    2024-04-01 08:24:08   |   2024-04-01 08:24:08
```

Examples

```sql
SELECT
    TO_DATE('2021-05-14') AS the_date,
    TO_TIME('06:13:00') AS the_time,
    TO_TIMESTAMP('2021-05-14T06:13:00') AS the_timestamp,

    -- Extract the DATE from a TIMESTAMP, we could do the same with TIME!
    transaction_timestamp::DATE AS casted_date
;
```

```
  the_date    |   the_time  |     the_timestamp     | casted_date
------------- | ----------- | --------------------- | ------------
 2021-05-14   |   06:13:00  | 2021-05-14T06:13:00   |  2021-05-14
```

Timezones

By default, `DATE` , `TIME` , and `TIMESTAMP` 's are stored without a timezone

`TIMESTAMP_NTZ`

- Default implementation of `TIMESTAMP`
- No timezone is stored

`TIMESTAMP_LTZ`

- Relies on **local** timezone

`TIMESTAMP_TZ`

- User **specifies** timezone

[1] https://docs.snowflake.com/en/sql-reference/data-types-datetime#timestamp-ltz-timestamp-ntz-timestamp-tz

Let's practice!