Changing Case

Changing Case

- Use the function UPPER() to return a string converted to uppercase
- Use the function LOWER() to return a string converted to lowercase
  - Case changes affect only letters. Digits, punctuation and whitespace are left unchanged
  - If argument is NULL, it returns NULL

Use an alias for a better output column name

```
SELECT UPPER(name), LOWER(name)
FROM members;
```

UPPER(name)	LOWER(name)
AILYN LUCEY	ailyn lucey
MILTIE CRACKEL	miltie crackel
ARDA BROWNLIE	arda brownlie
PORTIA ENNEVER	portia ennever

Concatenating Strings

Concatenating Strings with Concat()

- Use Concat() to concatenate stings
- The following applies:
  - Concatenation does NOT add space between strings
  - Concatenating a null involves a null
  - Concatenating a string and a nonstring, you must convert the nonstring to a string

```
SELECT CONCAT(first_name, ' ', last_name) AS Name
FROM customer_data;
```

Name
Mariya Bourget
Brooks Bewicke
Rikki Kerbey
Anya Jackson

Concatenating Strings with Concat()

- If your column contains nulls – you may get unexpected results

```
SELECT CONCAT(name,' ',suffix)
FROM members;
```

CONCAT(name,' ',suffix)
Miltie Crackel Sr

- Use the COALESCE function to handle any potential issues

```
SELECT CONCAT(name,' ',COALESCE(suffix, ''))
FROM members;
```

CONCAT(name,' ',COALESCE(suffix, ''))
Ailyn Lucey
Miltie Crackel Sr
Arda Brownlie

Concatenating Strings with Concat()

- You can use Concat() in SELECT, WHERE and ORDER BY clauses

```
SELECT *
FROM customer_data
WHERE CONCAT(first_name, ' ', last_name) = 'Abby Hulle';
```

id	first_name	last_name	email	city	country	address	state
20	Abby	Hulle	ahullej@nymag.com	Meridian	United States	0469 4th Center	MS

Using Mathematical Operations

Operators and Functions

- Allow you to calculate results derived from column values and other data
- You can perform the following:
  - Arithmetic operations – cut everyone's salary by a set %
  - String operations – concatenate first and last name
  - Datetime operations – compute the interval between two dates
  - System operations – find out what time the DBMS thinks it is

- ***Operator*** is a symbol or keyword indicating the operation that acts on one or more elements
- ***Operands*** are the SQL expressions
- ***Function*** is a built-in, named routine that performs a specialized task
  - Functions can take parenthesized ***arguments***

Arithmetic Operation

- Mathematical operations can be applied to results
- Arithmetic operators perform mathematical operation on two operands
- These include:
  - + for addition
  - - for subtraction
  - * for multiplication
  - / for division
  - -expr will reverse the sign of expr

Arithmetic Operation

The result of any arithmetic operation that involves a null is null

Use parenthesis to control the calculation order

When calculating multiple data types, the system will convert to the most complex type
(integer X decimal value will convert to a decimal value)

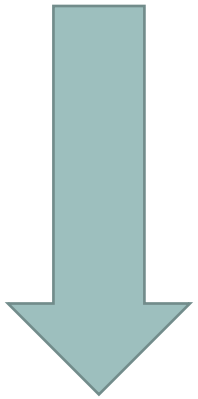
```
SELECT grocery, vendor, qty, buy_price, qty * buy_price AS totalSpent
FROM groceries;
```

grocery	vendor	qty	buy_price	totalSpent
Rice - Brown	Abshire and Sons	395	16.36	6462.20
Pepper - Cayenne	Kemmer, Lakin and Huels	1	5.51	5.51
Cake Circle, Foil, Scallop	Huels, Lehner and Purdy	398	7.24	2881.52
Lychee	Kilback Group	685	4.89	3349.65

Arithmetic Operations

```
SELECT grocery, vendor, qty, buy_price, qty * buy_price AS totalSpent
FROM groceries
ORDER BY totalSpent DESC;
```

grocery	vendor	qty	buy_price	totalSpent
Trueblue - Blueberry	Harvey, Kshlerin and Dickinson	827	19.60	16209.20
Carbonated Water - Strawberry	McGlynn, Macejkovic and Russel	934	15.19	14187.46
Cocoa Powder - Natural	Erdman Inc	762	14.93	11376.66
Pickerel - Fillets	Wintheiser and Sons	645	17.21	11100.45



Working with Substrings

Extracting a SUBSTRING()

- Use the function SUBSTRING() to extract a part of a string
  - A substring is any sequence of contiguous characters
  - Extracts part of a string starting at a specified position and continuing for a specified number of characters
  - SUBSTRING(string FROM start FOR LENGTH)

  You can use SUBSTRING() in SELECT, WHERE and ORDER BY

```
SELECT name, SUBSTRING(name FROM 1 FOR 1) AS firstLetter
FROM animalNames
ORDER BY firstLetter;
```

name	firstLetter
African elephant	A
Albatross, galapagos	A

Extracting a SUBSTRING()

```
SELECT name, SUBSTRING(name FROM 1 FOR 1) AS firstLetter
FROM animalNames
WHERE SUBSTRING(name FROM 1 FOR 1) = 'B';
```

name	firstLetter
Bee-eater, rainbow	B
Brush-tailed rat kangaroo	B
Bee-eater, carmine	B
Blackbird, red-winged	B
Bettong, brush-tailed	B
Blesbok	B
Boubou, southern	B
Butterfly (unidentified)	B

Extracting a SUBSTRING()

```
SELECT name, SUBSTRING(name FROM 2)
FROM animalNames;
```

name	SUBSTRING(name FROM 2)
African elephant	frican elephant
Northern phalarope	orthern phalarope
Sloth, two-toed tree	loth, two-toed tree

Word Length

String Length

- CHARACTER_LENGTH() is used to return the number of characters in a string
- Returns an integer greater than or equal to 0
- Length of an empty string is zero

```
SELECT name, CHARACTER_LENGTH(name) AS nameLength
FROM animalNames
ORDER BY nameLength DESC;
```

name	nameLength
Darwin ground finch (unidentified)	34
Hornbill, leadbeateri's ground	30
Otter, oriental short-clawed	28

String Length

```
SELECT name, CHARACTER_LENGTH(name) AS nameLength
FROM animalNames
WHERE CHARACTER_LENGTH(name) < 10
ORDER BY nameLength DESC;
```

name	nameLength
Steenbok	8
Steenbok	8
Ox, musk	8
Blesbok	7

Removing Extra Characters

Trimming with TRIM()

- Use the TRIM() function to remove unwanted characters from the ends of strings
  - It removes leading characters, trailing characters, or both
  - Trims spaces by default, but you can use to strip out any unwanted characters
  - Used for trimming trailing spaces from CHAR values
  - Trimming has no effect on empty strings

```
SELECT name, CHARACTER_LENGTH(name) AS nameLength, TRIM(LEADING 'S' FROM name)
FROM animalNames
WHERE SUBSTRING(name FROM 1 FOR 1) = 'S'
ORDER BY nameLength DESC;
```

name	nameLength	TRIM(LEADING 'S' FROM name)
Smith's bush squirrel	21	mith's bush squirrel
Sloth, two-toed tree	20	loth, two-toed tree

Trimming with TRIM()

```
SELECT name, CHARACTER_LENGTH(name) AS nameLength, TRIM(TRAILING 'k' FROM name)
FROM animalNames
WHERE SUBSTRING(name FROM 1 FOR 1) = 'S'
ORDER BY nameLength DESC;
```

Steenbok	8	Steenbo
Steenbok	8	Steenbo
Sambar	6	Sambar

Getting Position of Information

POSITION()

- POSITION() returns an integer value
- Returns 0 for no substring match
- POSITION(substring IN string)
- Where substring is the string to search for and string is the string to search.
- Each argument is a string expressions such as a column that contains character strings

```
SELECT name, POSITION('l' IN name)
FROM animalNames
```

name	POSITION('l' IN name)
African elephant	10
Northern phalarope	13

POSITION

```
SELECT name, POSITION('z' IN name)
FROM animalNames;
```

name	POSITION('z' IN name)
African elephant	0
Northern phalarope	0
Sloth, two-toed tree	0
Blue-tongued lizard	16

POSITION()

```
SELECT name, POSITION('Afr' IN name)
FROM animalNames;
```

name	POSITION('Afr' IN name)
African elephant	1
Northern phalarope	0
Sloth, two-toed tree	0

POSITION()

```
SELECT name, POSITION('a' IN name)
FROM animalNames;
```

name	POSITION('a' IN name)
African elephant	1
Northern phalarope	12
Sloth, two-toed tree	0