# Reporting with SQL

## Ordering, Limiting, and Paging Results

### Overview

Handling Data

Ways to retrieve data:

* Ordering data
* Limiting data results, e.g., latest data only
* Manipulating text, e.g., upper/lower case
* Aggregation, e.g., sum, min, max
* Dates

### Retrieving Results in a Particular Order

Ordering by a single column criteria:

SELECT \* FROM *<table name>* ORDER BY *<column>* [ASC|DESC];

Ordering by multiple column criteria:

SELECT \* FROM <table name> ORDER BY *<column>* [ASC|DESC],

*<column 2>* [ASC|DESC],

...,

*<column n>* [ASC|DESC];

### Limiting the Number of Results

#### SQLite, PostgreSQL and MySQL

To limit the number of results returned, use the LIMIT keyword.

SELECT *<columns>* FROM *<table>* LIMIT *<# of rows>*;

#### MS SQL

To limit the number of results returned, use the TOP keyword.

SELECT TOP *<# of rows> <columns>* FROM *<table>*;

#### Oracle

To limit the number of results returned, use the ROWNUM keyword in a WHERE clause.

SELECT *<columns>* FROM *<table>* WHERE ROWNUM <= *<# of rows>*;

### Paging Through Results

#### SQLite, PostgreSQL and MySQL

To page through results you can either use the OFFSET keyword in conjunction with the LIMIT keyword or just with LIMIT alone.

SELECT *<columns>* FROM *<table>* LIMIT *<# of rows>* OFFSET *<skipped rows>*;

SELECT *<columns>* FROM *<table>* LIMIT *<skipped rows>*, *<# of rows>*;

#### MS SQL and Oracle

To page through results you can either use the OFFSET keyword in conjunction with the FETCH keyword. Cannot be used with TOP.

SELECT *<columns>* FROM *<table>* OFFSET *<skipped rows>* ROWS

FETCH NEXT *<# of rows>* ROWS ONLY;

#### Quiz

* Default sort order: ASC
* Example:

SELECT TOP 50 name FROM people;

## Working with Text

### What are Functions?

Syntax:

* Keywords: data presented unaltered
* Operators: comparisons and simple manipulation
* Functions: presents data differently through manipulation
  + SUM()
  + UPPER()

### Adding Text Columns Together

#### SQLite, PostgreSQL and Oracle

Use the concatenation operator ||.

SELECT <value or column> || <value or column> || <value or column> FROM <table>;

#### MS SQL

Use the concatenation operator +.

SELECT <value or column> + <value or column> + <value or column> FROM <table>;

#### MySQL, Postgres and MS SQL

Use the CONCAT() function.

SELECT CONCAT(<value or column>, <value or column>, <value or column>) FROM <table>;

### Single vs Double Quotes

In SQL there's a difference between using single quotes (') and double quotes ("). Single quotes should be used for String literals (e.g. 'lbs'), and double quotes should be used for identifiers like column aliases (e.g. "Max Weight"):

SELECT maximum\_weight || 'lbs' AS "Max Weight" FROM ELEVATOR\_DATA;

### Finding the Length of Text

To obtain the length of a value or column use the LENGTH() function.

SELECT LENGTH(<value or column>) FROM <tables>;

### Changing the Case of Text Columns

Use the UPPER() function to uppercase text.

SELECT UPPER(<value or column>) FROM <table>;

Use the LOWER() function to lowercase text.

SELECT LOWER(<value or column>) FROM <table>;

### Creating Excerpts from Text

To create smaller strings from larger piece of text you can use the SUBSTR() funciton or the substring function.

SELECT SUBSTR(<value or column>, <start>, <length>) FROM <table>;

### Replacing Portions of Text

To replace piece of strings of text in a larger body of text you can use the REPLACE() function.

SELECT REPLACE(<original value or column>, <target string>, <replacement string>) FROM <table>;