```
--------------------------------------------------------------------------------
-- Basic SELECT Information
--------------------------------------------------------------------------------
-- -- indicates a comment - anything following on the line is ignored
--
-- SQL SELECT statement - retrieve values form the database (Read)
--
-- A SELECT statement is often referred to as a query
--
-- Basic syntax:
--
--      SELECT   - columns to include in the result (separate mutiple column requests with commas)
--      FROM     - table containing rows with columns used in the query
--      WHERE    - rows to include in the result (row filter)
--      ORDER BY - sequence of rows in the result
--                 without an ORDER BY the sequence of the rows in the result is not predictable
--                 if the sequence of the rows in the result matter - code an ORDER BY
--
-- WHERE predicates (condition):
--
--  not equals is either  <>   or     !=
--
--          =  <>  !=  >  >=  <  <=
--          IN(list-of-values)       -- alterative to a series of = OR
--          NOT IN(list-of-values)   -- alterative to a series of != AND
--          BETWEEN value AND value
--          IS NULL       = NULL does not work - NULL is unknown value
--          IS NOT NULL  != NULL and <> NULL does not work - NULL is unknown value
--          LIKE    (use wildcards: % means 0 to any number of any characters
--                               _ means exactly any one character)
--                'value%'  - starts with the value
--                '%value'  - ends with the value
--                '%value%' - contains the value

          LIKE does a case-insenitive search by default - mySQL extension

--
--          LIKE BINARY  (case sensitive LIKE - mySQL extension)
--
-- predicates may be combined using AND and OR - full predicates must be coded on both sides for AND / OR
--
--   column-name predicate column-name/value OR column-name predicate column-name/value
--
-- use parentheses to make your multi-predicate condition clear

-- The DISTINCT clause on a SELECT removes duplicate values from the result
-- based on the all columns that follow
--
----------------------------------------------------------------
```