/\*

In SQL, you can use the LIKE command in order to search through text-based values. With LIKE, there are two special characters: % and \_.  
  
The percent sign (%) represents zero, one, or multiple characters.  
  
The underscore (\_) represents one character.

***robots***

|  |  |
| --- | --- |
| **id** | **name** |
| 1 | Robot 2000 |
| 2 | Champion Robot 2001 |
| 3 | Dragon |
| 4 | Turbo Robot 2002 |
| 5 | Super Robot 2003 |
| 6 | Super Turbo Robot 2004 |
| 7 | Not A Robot |
| 8 | Unreleased Turbo Robot 2111 |

Consider the above table

Can you run a query that returns "Robot" followed by a year between 2000 and 2099? (So 2015 is a valid value at the end, but 2123 is not.)

\*/

SELECT \* FROM robots WHERE name LIKE "%Robot 20%" ;

/\*

You can use a CASE statement to return certain values when certain scenarios are true.

Can you return the results with a column named sound that returns "talk" for humans, "bark" for dogs, and "meow" for cats?

***friends\_of\_pickles***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **id** | **name** | **gender** | **species** | **height\_cm** |
| 1 | Dave | male | human | 180 |
| 2 | Mary | female | human | 160 |
| 3 | Fry | male | cat | 30 |
| 4 | Leela | female | cat | 25 |
| 5 | Odie | male | dog | 40 |
| 6 | Jumpy | male | dog | 35 |
| 7 | Sneakers | male | dog | 55 |

Consider the above table

\*/

SELECT \*,

CASE WHEN species = 'human' THEN 'talk'

WHEN species = 'dog' THEN 'bark'

ELSE 'meow'

END AS sound

FROM friends\_of\_pickles;

/\*

In SQL, you can search for the substring of a given value. Perhaps a location is stored in the format "city, state" and you just want to grab the state.  
  
SUBSTR is used in this format: SUBSTR(column\_name, index, number\_of\_characters)  
  
index is a number that denotes where you would start the substring. 1 would indicate the first character, 2 would indicated the second character, etc.

Here are some examples:  
SUBSTR(name, 1, 5) is the first 5 characters of the name.  
SUBSTR(name, -4) is the last 4 characters of the name.

In other versions of SQL, you could use RIGHT to do this.

Can you return all of the robots that are located in NY?

***robots***

|  |  |  |
| --- | --- | --- |
| **id** | **name** | **location** |
| 1 | R2000 - Robot 2000 | New City, NY |
| 2 | R2001 - Champion Robot 2001 | Palo Alto, CA |
| 3 | D0001 - Dragon | New York City, NY |
| 4 | R2002 - Turbo Robot 2002 | Spring Valley, NY |
| 5 | R2003 - Super Robot 2003 | Nyack, NY |
| 6 | R2004 - Super Turbo Robot 2004 | Tampa, FL |
| 7 | N0001 - Not A Robot | Seattle, WA |
| 8 | U2111 - Unreleased Turbo Robot 2111 | Buffalo, NY |

Consider the above table  
\*/

SELECT \* FROM robots WHERE SUBSTR(location,-2) LIKE 'NY';