

# Selecting Data

The **select** statement is used to query the database and retrieve selected data that match the criteria that you specify. Here is the format of a simple select statement:

```
select "column1"  
      [, "column2", etc]  
from "tablename"  
[where "condition"];  
[] = optional
```

The column names that follow the select keyword determine which columns will be returned in the results. You can select as many column names that you'd like, or you can use a "\*" to select all columns.

The table name that follows the keyword **from** specifies the table that will be queried to retrieve the desired results.

The **where** clause (optional) specifies which data values or rows will be returned or displayed, based on the criteria described after the keyword **where**.

Conditional selections used in the **where** clause:

- = Equal
- > Greater than
- < Less than
- >= Greater than or equal
- <= Less than or equal
- <> Not equal to
- LIKE \*See note below

The **LIKE** pattern matching operator can also be used in the conditional selection of the where clause. Like is a very powerful operator that allows you to select only rows that are "like" what you specify. The percent sign "%" can be used as a wild card to match any possible character that might appear before or after the characters specified. For example:

```
select first, last, city  
from empinfo  
where first LIKE 'Er%';
```

This SQL statement will match any first names that start with 'Er'. **Strings must be in single quotes.**

Or you can specify,

```
select first, last  
from empinfo  
where last LIKE '%s';
```

This statement will match any last names that end in a 's'.

```
select * from empinfo  
where first = 'Eric';
```

This will only select rows where the first name equals 'Eric' exactly.

| Sample Table: empinfo |          |       |     |            |            |
|-----------------------|----------|-------|-----|------------|------------|
| first                 | last     | id    | age | city       | state      |
| John                  | Jones    | 99980 | 45  | Payson     | Arizona    |
| Mary                  | Jones    | 99982 | 25  | Payson     | Arizona    |
| Eric                  | Edwards  | 88232 | 32  | San Diego  | California |
| Mary Ann              | Edwards  | 88233 | 32  | Phoenix    | Arizona    |
| Ginger                | Howell   | 98002 | 42  | Cottonwood | Arizona    |
| Sebastian             | Smith    | 92001 | 23  | Gila Bend  | Arizona    |
| Gus                   | Gray     | 22322 | 35  | Bagdad     | Arizona    |
| Mary Ann              | May      | 32326 | 52  | Tucson     | Arizona    |
| Erica                 | Williams | 32327 | 60  | Show Low   | Arizona    |
| Leroy                 | Brown    | 32380 | 22  | Pinetop    | Arizona    |
| Elroy                 | Cleaver  | 32382 | 22  | Globe      | Arizona    |

Enter the following sample select statements in the SQL Interpreter Form at the bottom of this page. Before you press "submit", write down your expected results. Press "submit", and compare the results.

```
select first, last, city from empinfo;

select last, city, age from empinfo
where age > 30;

select first, last, city, state from empinfo
where first LIKE 'J%';

select * from empinfo;

select first, last, from empinfo
where last LIKE '%s';

select first, last, age from empinfo
where last LIKE '%illia%';

select * from empinfo where first = 'Eric';
```

## Select statement exercises

Enter select statements to:

1. Display the first name and age for everyone that's in the table.
2. Display the first name, last name, and city for everyone that's not from Payson.
3. Display all columns for everyone that is over 40 years old.
4. Display the first and last names for everyone whose last name ends in an "ay".
5. Display all columns for everyone whose first name equals "Mary".
6. Display all columns for everyone whose first name contains "Mary".

