```
SELECT drink_name FROM easy_drinks
WHERE main IN ('peach nectar', 'soda');
```

```
SELECT drink_name FROM easy_drinks
WHERE main BETWEEN 'P' AND 'T';
```

```
SELECT drink_name FROM drink_info
WHERE NOT calories = 0;
```

```
SELECT drink_name FROM drink_info
WHERE calories > 0;
```

```
SELECT drink_name FROM drink_info
WHERE NOT carbs BETWEEN 3 AND 5;
```

```
SELECT drink_name FROM drink_info
WHERE carbs < 3
OR
carbs > 5;
```

```
SELECT date_name from black_book
WHERE NOT date_name LIKE 'A%'
AND NOT date_name LIKE 'B%';
```

```
SELECT date_name FROM black_book
WHERE date_name NOT BETWEEN 'A' AND 'C';
```

Column Name	Description	Example	Best Choice of Data Type
price	The cost of an item for sale	5678.39	DEC(6,2)
zip_code	4 number zip code	2919	CHAR(4)
atomic_weight	Atomic weight of an element with up to 6 decimal places	1.052965	DEC(10,6)
comments	Large block of text, more than 255 characters	Joe, I'm at the shareholder's meeting. They just gave a demo and there were rubber duckies flying around the screen. Was this your idea of a joke? You might want to spend some time on Monster.com.	BLOB
quantity	How many of this item in stock	255	INT
tax_rate	up to 3 dec	3.755	DEC(6,3)
book_title	text string	Head First SQL	VARCHAR(50)
gender	One character, either M or F	M	CHAR(1)
phone_number	Ten digits, no punctuation	2105552367	CHAR(10)
state	Two-character abbreviation for a state	TX, CA	CHAR(2)
anniversary	DAY, MONTH, YEAR	11/22/2006	DATE
games_won	INT representing # won	15	INT
meeting_time	A TIME	10:30 a.m. 4/12/2020	DATETIME

Sharpen your pencil

Your SQL RDBMS will tell you when something is wrong with your statement, but will sometimes be a bit vague. Take a look at each INSERT statement below. First try to guess what's wrong with the statement, and then try typing it in to see what your RDBMS reports.

```
INSERT INTO my_contacts

(last_name, first_name, email, gender, birthday, profession, location, status,
interests, seeking) VALUES ('Anderson', 'Jillian', 'jill_anderson@breakneckpizza.net',
'F', '1980-09-05', 'Technical Writer', 'Single', 'Kayaking, Reptiles', 'Relationship,
Friends');
```

What's wrong? No location value

Your RDBMS says: ERROR 1136 (21S01): Column count doesn't match value count at row 1

```
INSERT INTO my_contacts

(last_name, first_name, gender, birthday, profession, location, status, interests,
seeking) VALUES ('Anderson', 'Jillian', 'jill_anderson@breakneckpizza.net', 'F',
'1980-09-05', 'Technical Writer', 'Palo Alto, CA', 'Single', 'Kayaking, Reptiles',
'Relationship, Friends');
```

What's wrong? missing email column

Your RDBMS says: a ERROR 1136 (21S01): Column count doesn't match value count at row 1

```
INSERT INTO my_contacts

(last_name, first_name, email, gender, birthday, profession, location, status,
interests, seeking) VALUES ('Anderson', 'Jillian', 'jill_anderson@breakneckpizza.net',
'F', '1980-09-05', 'Technical Writer' 'Palo Alto, CA', 'Single', 'Kayaking, Reptiles',
'Relationship, Friends');
```

What's wrong? missing comma between 'technical writer' and 'palo atto'

Your RDBMS says: ERROR 1136 (21S01): Column count doesn't match value count at row 1

```
INSERT INTO my_contacts

(last_name, first_name, email, gender, birthday, profession, location, status,
interests, seeking) VALUES ('Anderson', 'Jillian', 'jill_anderson@breakneckpizza.net',
'F', '1980-09-05', 'Technical Writer', 'Palo Alto, CA', 'Single', 'Kayaking, Reptiles',
'Relationship, Friends);
```

What's wrong? missing single quote at end

If this one causes your RDBMS to "hang" try typing a single quote followed by a semicolon after you've entered the rest of the statement.

Your RDBMS says:
Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near "Relationship, Friends)' at line 5

If you have data in your table with quotes, you might actually have to search for it with a WHERE clause at some point. To SELECT data containing single quotes in your WHERE clause, you need to escape your single quote, just like you did when you inserted it.

Rewrite the code below using the different methods of escaping the single quote.

```
SELECT * FROM my_contacts
WHERE
location = 'Grover's Mill, NJ';
```

1

```
SELECT * FROM my_contacts
WHERE
location = 'Grover''s Mill, NJ';
```

2

```
SELECT * FROM my_contacts
WHERE
location = 'Grover\'s Mill, NJ';
```

Which method do you prefer?

```
SELECT drink_name FROM easy_drinks
WHERE main = 'cherry juice';

SELECT drink_name FROM easy_drinks
WHERE second = 'apricot nectar';

SELECT drink_name FROM easy_drinks
WHERE amount1 = 2;

SELECT drink_name FROM easy_drinks
WHERE amount2 = 7;
```

Now write three SELECT statements that will give you a Bull Frog.

1.
```
SELECT drink_name FROM easy_drinks
WHERE main = 'iced tea';
```

2.
```
SELECT drink_name FROM easy_drinks
WHERE second = 'lemonade';
```

3.
```
SELECT drink_name FROM easy_drinks
WHERE amount1 = 1.5;
```

So I could have found Anne using AND?

Using the my_contacts table, write some queries for Greg. SELECT only the columns you really need to give you your answer. Pay attention to single quotes.

Write a query to find the email addresses of all computer programmers.

```
SELECT email FROM my_contacts
WHERE profession = 'computer programmers';
```

Write a query to find the name and location of anyone with your birthdate.

```
SELECT last_name, first_name, location
FROM my_contacts
WHERE birthday = '1975-09-05'
```

Write a query to find the name and email of any single people who live in your town. For extra points, only pick those of the gender you'd want to date.

```
SELECT last_name, first_name, email
FROM my_contacts
WHERE location = 'San Antonio, TX'
AND gender = 'M'
AND status = 'single'
```

Write the query Greg could have used to find all the Annes from San Francisco.

```
SELECT last_name, first_name, email
FROM my_contacts
WHERE location = 'San Fran, CA'
AND first_name = 'Anne';
```

Sharpen your pencil

Your turn to do some mixing. Write queries that will return the following information. Also write down what the result of each query is:

The cost of each drink with ice that is yellow and has more than 33 calories.

```
SELECT cost FROM drink_info
WHERE ice = 'Y'
AND color = 'yellow'
AND calories > 33;
```

Result: 14.00

The name and color of each drink which does not contain more than 4 grams of carbs and uses ice.

```
SELECT drink_name, color FROM drink_info
WHERE ice = 'Y'
AND carbs <= 4;
```

Result: Blue moon blue

The cost of each drink whose calorie count is 80 or more.

```
SELECT cost FROM drink_info
WHERE calories >= 80;
```

Result: 5.50, 3.20, 2.60

Drinks called Greyhound and Kiss on the Lips, along with each one's color and whether ice is used to mix the drink, without using the names of the drinks in your query.

```
SELECT drink_name, color, ice FROM drink
WHERE cost > 3.0;
```

Result: Kiss on the lips, purple, y, greyhound, yellow, y

Sharpen your pencil

Cross out the unnecessary parts of the two SELECTs below and add an OR to turn it into a single SELECT statement.

```
SELECT drink_name FROM easy_drinks WHERE
main = 'orange juice';


SELECT drink_name FROM easy_drinks WHERE
main = 'apple juice';
```

Use your new selection skills to rewrite your new SELECT.

```
SELECT drink_name FROM easy_drinks
WHERE
main = 'orange juice'
OR
main = 'apple juice'
```

Sharpen your pencil

Rewrite the query on the previous page to SELECT all the names of drinks that have more than 60 calories and less than 30.

```
SELECT drink_name FROM drink_info
WHERE
calories > 60
AND
calories < 30
```

Try using BETWEEN on text columns. Write a query that will SELECT the names of drinks that begin with the letters G through O.

```
SELECT drink_name FROM drink_info
WHERE
drink_name BETWEEN 'G' AND 'P';
```

What do you think the results of this query will be?

```
SELECT drink_name FROM drink_info WHERE
calories BETWEEN 60 AND 30;
```

error

Rewrite each of the following WHERE clauses so they are as simple as possible. You can use AND, OR, NOT, BETWEEN, LIKE, IN, IS NULL, and the comparison operators to help you. Refer back to the tables used in this chapter.

```
SELECT drink_name from easy_drinks
WHERE NOT amount1 < 1.50;
```

```
SELECT drink_name FROM easy_drinks
WHERE amount1 >= 1.50;
```

```
SELECT drink_name FROM drink_info
WHERE NOT ice = 'Y';
```

```
SELECT drink_name FROM drink_info
WHERE ice = 'N';
```

```
SELECT drink_name FROM drink_info
WHERE NOT calories < 20;
```

```
SELECT drink_name FROM drink_info
WHERE calories >= 20;
```