INSERT

Adding Data to Your Tables

INSERT

```
INSERT INTO cats(name, age)
VALUES ('Jetson', 7);
```

```
INSERT INTO cats(name, age)
VALUES ("Jetson", 7);
```

```
INSERT INTO cats(name, age) VALUES ("Jetson", 7);
```

```
INSERT INTO cats
            (NAME,
             age)
VALUES      ("jetson",
             7);
```

THE ORDER MATTERS

```sql
INSERT INTO cats(age, name)
VALUES (12, 'Victoria');
```

Let's Try It!

So… How Do We Know It Worked?

Sometimes there is no easy order to teach this stuff... Here's a command from the next section:

```
SELECT * FROM cats;
```

MULTIPLE INSERT

```sql
INSERT INTO cats(name, age)
VALUES ('Charlie', 10)
      ,('Sadie', 3)
      ,('Lazy Bear', 1);
```

Time For You To Try!

Create a *people* table

- first_name - 20 char limit
- last_name - 20 char limit
- age

Insert Your 1st Person!

first_name	last_name	age
'Tina'	'Belcher'	13

Insert Your 2nd Person!

first_name	last_name	age
'Bob'	'Belcher'	42

Multiple Insert Time!

first_name	last_name	age
'Linda'	'Belcher'	45
'Phillip'	'Frond'	38
'Calvin'	'Fischoeder'	70

To view errors and warnings

```
SHOW WARNINGS;
```

What's Up With This?

```
+--------+------------+------+-----+---------+-------+
| Field  | Type       | Null | Key | Default | Extra |
+--------+------------+------+-----+---------+-------+
| name   | varchar(5) | YES  |     | NULL    |       |
| age    | int(11)    | YES  |     | NULL    |       |
+--------+------------+------+-----+---------+-------+
```

"The Value Is Not Known"

Null Does Not Mean Zero!

Right now, we could do this...

```sql
INSERT INTO cats(name)
VALUES ('Bean');
```

Bean is a great name for a cat

Or This! *gasp*

```
INSERT INTO cats()
VALUES ();
```

The Solution?

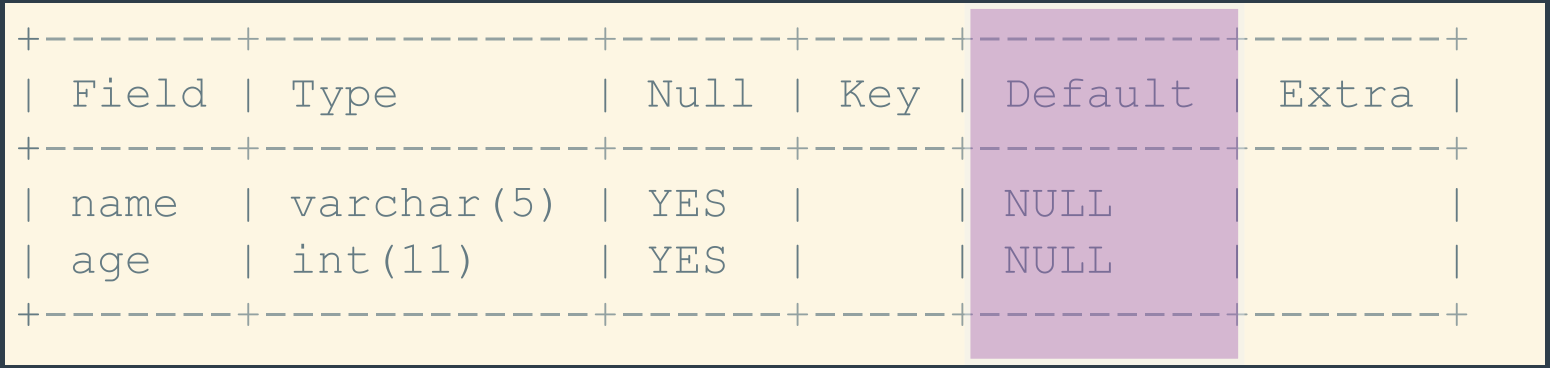
NOT NULL

```
CREATE TABLE cats2
  (
    name VARCHAR(100) NOT NULL,
    age  INT NOT NULL
  );
```

Notice The Difference!

```
+-------+--------------+------+-----+---------+-------+
| Field | Type         | Null | Key | Default | Extra |
+-------+--------------+------+-----+---------+-------+
| name  | varchar(100) | NO   |     | NULL    |       |
| age   | int(11)      | NO   |     | NULL    |       |
+-------+--------------+------+-----+---------+-------+
```

What's Up With This?



```
+-------+------------+------+-----+---------+-------+
| Field | Type       | Null | Key | Default | Extra |
+-------+------------+------+-----+---------+-------+
| name  | varchar(5) | YES  |     | NULL    |       |
| age   | int(11)    | YES  |     | NULL    |       |
+-------+------------+------+-----+---------+-------+
```

To Set Default Values

```
CREATE TABLE cats3
  (
    name VARCHAR(100) DEFAULT 'unnamed',
    age  INT DEFAULT 99
  );
```

Isn't This Redundant?

```sql
CREATE TABLE cats4
  (
    name VARCHAR(100) NOT NULL DEFAULT 'unnamed',
    age  INT NOT NULL DEFAULT 99
  );
```

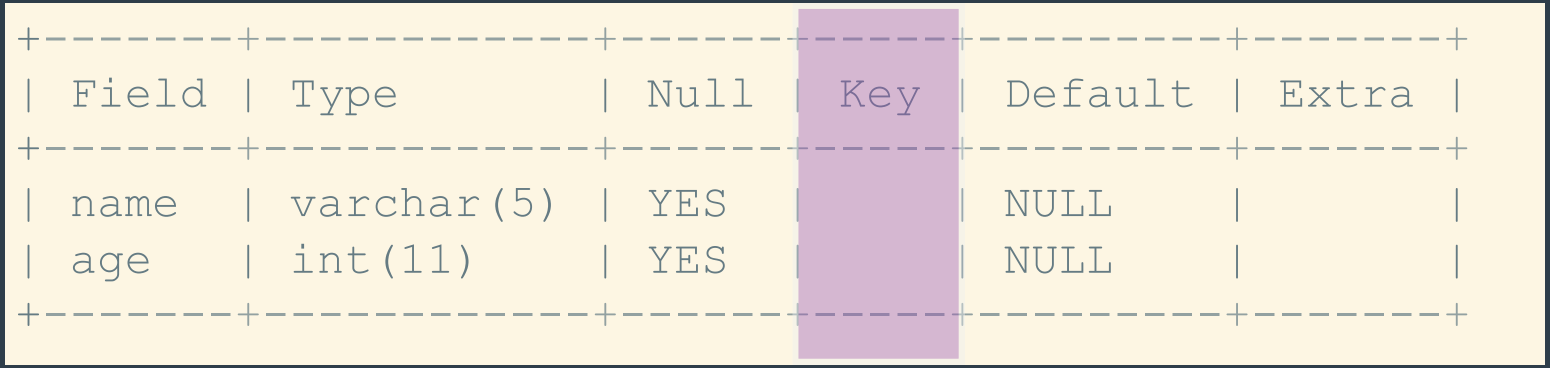
No!

We can still manually set things to NULL if we don't specify NOT NULL

```
INSERT INTO cats3(name, age)
VALUES(NULL, 3);
```

One More Thing

What's Up With This?


```
+-------+------------+------+-----+---------+-------+
| Field | Type       | Null | Key | Default | Extra |
+-------+------------+------+-----+---------+-------+
| name  | varchar(5) | YES  |     | NULL    |       |
| age   | int(11)    | YES  |     | NULL    |       |
+-------+------------+------+-----+---------+-------+
```

Right now, this could happen!

Name	Breed	Age
Monty	Tabby	10
Monty	Tabby	10
Monty	Tabby	10
Monty	Tabby	10

How Do We Make Each Unique?

Name	Breed	Age	CatID
Monty	Tabby	10	1
Monty	Tabby	10	2
Monty	Tabby	10	3
Monty	Tabby	10	4

Primary Key

A Unique Identifier

Primary Key

```
CREATE TABLE unique_cats (
  cat_id INT NOT NULL PRIMARY KEY,
  name VARCHAR(100),
  age INT
 );
```

Another Option

```
CREATE TABLE unique_cats (
  cat_id INT NOT NULL,
  name VARCHAR(100),
  age INT,
  PRIMARY KEY(cat_id)
 );
```

NOT NULL is redundant

```
CREATE TABLE unique_cats (
  cat_id INT,
  name VARCHAR(100),
  age INT,
  PRIMARY KEY(cat_id)
 );
```

PRIMARY KEYs cannot be NULL

Auto-Increment

```
CREATE TABLE unique_cats3 (
    cat_id INT AUTO_INCREMENT,
    name VARCHAR(100),
    age INT,
    PRIMARY KEY (cat_id)
);
```

cat_id will automatically increment for each new cat inserted into the table

YOUR
TURN

Define an Employees table, with the following fields:

- id - number(automatically increments) and primary key
- last_name - text, mandatory
- first_name - text, mandatory
- middle_name - text, not mandatory
- age - number mandatory
- current_status - text, mandatory, defaults to 'employed'

THE SOLUTION

```
CREATE TABLE employees (
    id INT AUTO_INCREMENT PRIMARY KEY,
    last_name VARCHAR(100) NOT NULL,
    first_name VARCHAR(100) NOT NULL,
    middle_name VARCHAR(100),
    age INT NOT NULL,
    current_status VARCHAR(100) NOT NULL DEFAULT 'employed'
);
```