SQL DDL DML

DDL – Data Definition Language -- Modify Database Structures

- `Create`
- `Alter`
- `Drop`

DML – Data Manipulation Language -- Modify data values

- `Insert`
- `Update`
- `Delete`
- `Truncate`

SQL Insert

INSERT statement – Two Formats

```
1. INSERT INTO <table name> (column, column, column)
         VALUES (value, value, value)
```

(if no column & value are specified, NULL or default will be assigned)

```
2. INSERT INTO <table name>
         VALUES (value, value, value, value)
```

(must have a value or NULL for every column in the table)

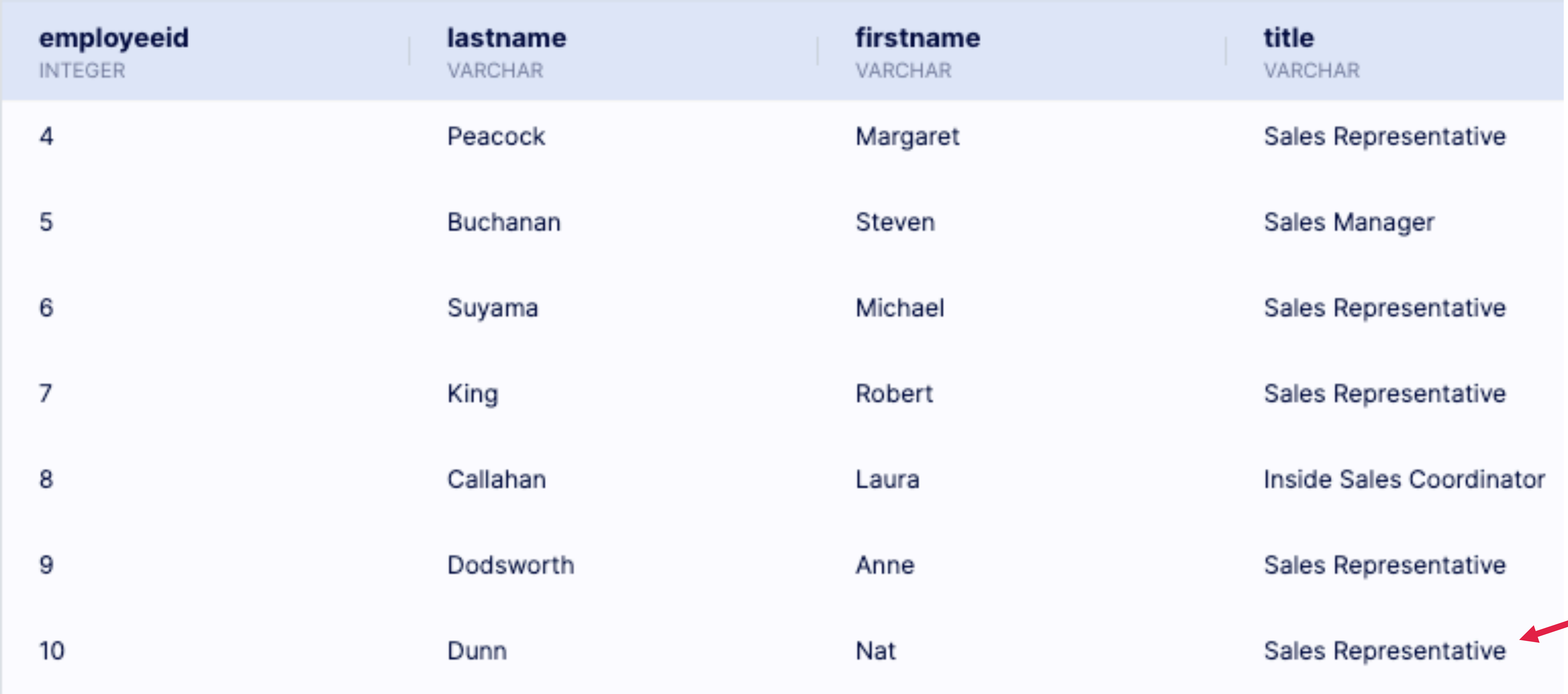
SQL Insert

Example: INSERT statement – Format 1

Add a new row to the employees table

```
INSERT INTO "alanparadise/nw"."employees"
    (employeeid, LastName, FirstName, Title, TitleOfCourtesy, BirthDate, HireDate,
     Address, City, Region, PostalCode, Country, HomePhone, Extension)
 VALUES
    (10,'Dunn','Nat','Sales Representative', 'Mr.', '1970-02-19', '2014-01-15',
    '4933 Jamesville Rd.','Jamesville','NY', '13078','USA',
    '315-555-5555','130');
```

SQL Insert



employeeid INTEGER	lastname VARCHAR	firstname VARCHAR	title VARCHAR
4	Peacock	Margaret	Sales Representative
5	Buchanan	Steven	Sales Manager
6	Suyama	Michael	Sales Representative
7	King	Robert	Sales Representative
8	Callahan	Laura	Inside Sales Coordinator
9	Dodsworth	Anne	Sales Representative
10	Dunn	Nat	Sales Representative

New Row

SQL Insert

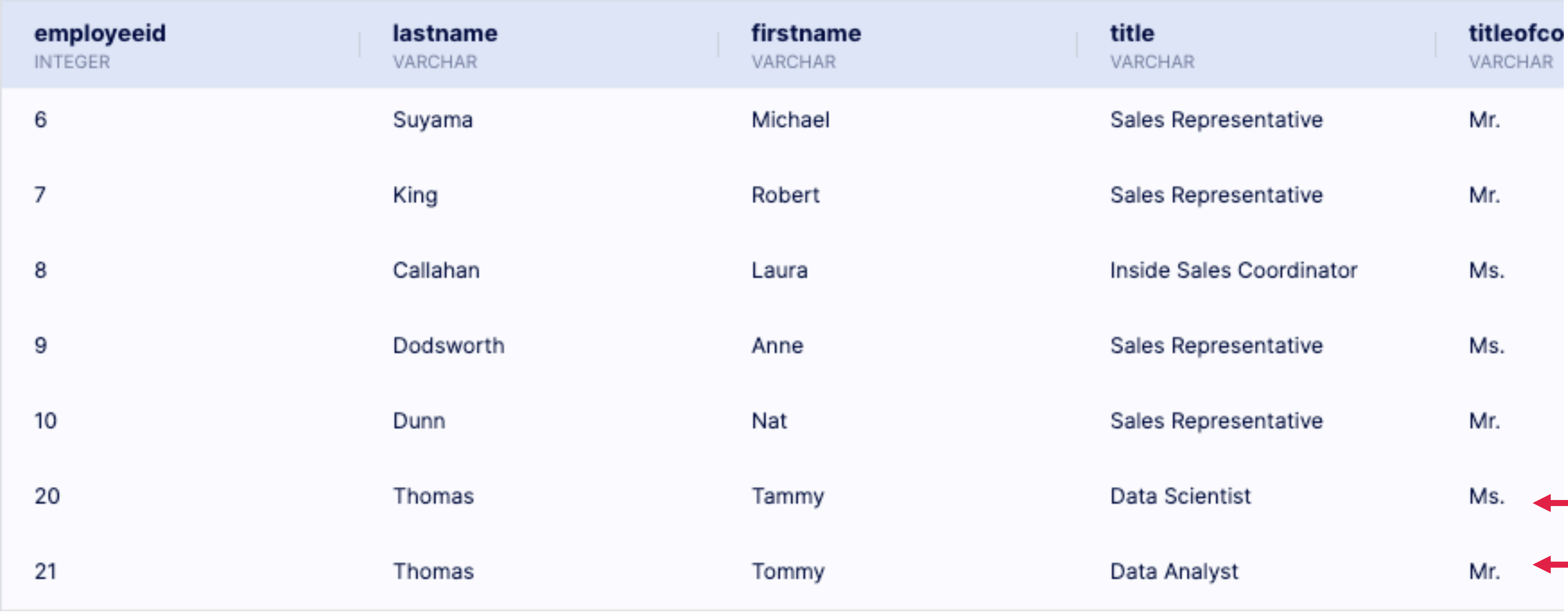
INSERT statement (Format 2 -- missing values)

Add a new row to the employees table

```
INSERT INTO "alanparadise/nw"."employees"
    VALUES
    ('20','Thomas','Tammy','Data Scientist','Ms.','1990-08-27',
     '2017-06-18', '5012 Arapahoe St.','Boulder','CO', '80304','USA');

INSERT INTO "alanparadise/nw"."employees"
    VALUES
    ('21','Thomas','Tommy','Data Analyst', 'Mr.','1991-09-28', '2017-06-18',
    '5014 Arapahoe St.','Boulder','CO', '80304','USA',NULL,NULL,NULL,NULL,NULL);
```

SQL Insert



employeeid	lastname	firstname	title	titleofco
INTEGER	VARCHAR	VARCHAR	VARCHAR	VARCHAR
6	Suyama	Michael	Sales Representative	Mr.
7	King	Robert	Sales Representative	Mr.
8	Callahan	Laura	Inside Sales Coordinator	Ms.
9	Dodsworth	Anne	Sales Representative	Ms.
10	Dunn	Nat	Sales Representative	Mr.
20	Thomas	Tammy	Data Scientist	Ms.
21	Thomas	Tommy	Data Analyst	Mr.

New Rows

SQL – Update

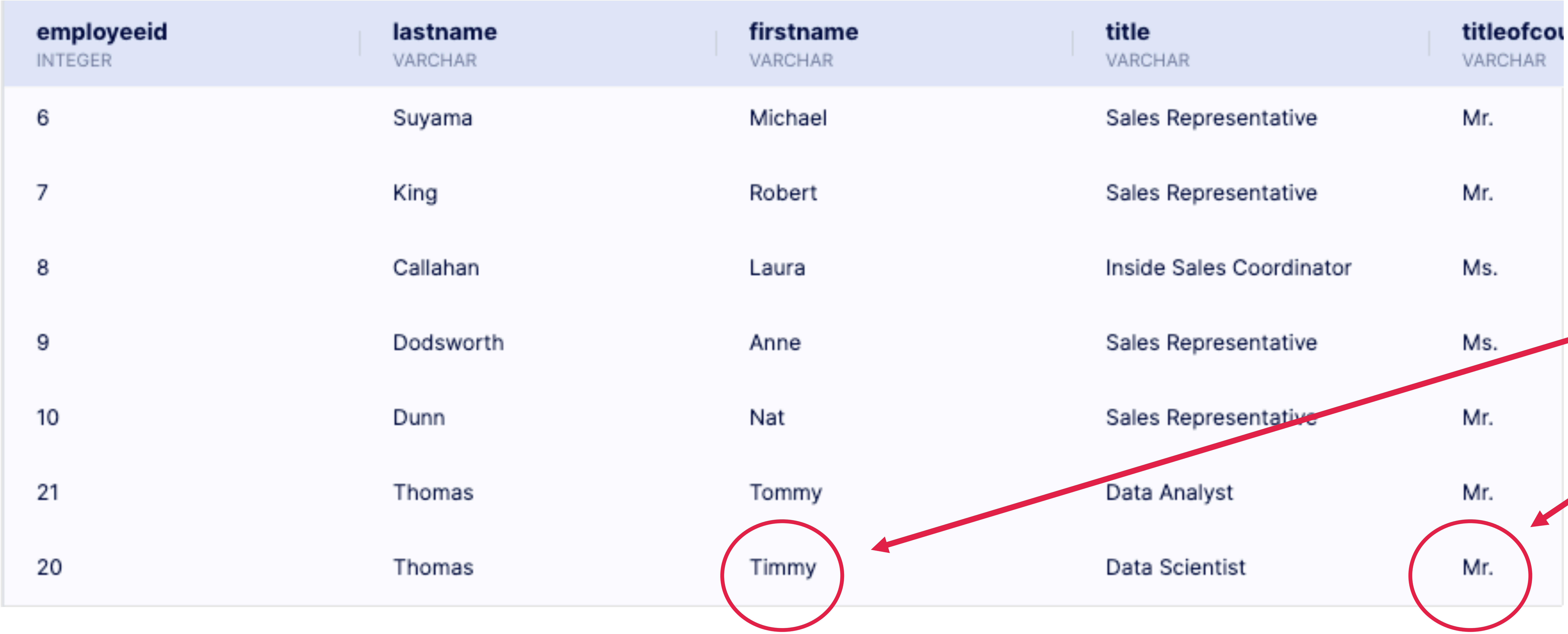
UPDATE statement – assigns new values to specific columns based on a condition

```
  UPDATE <table name>
      SET <column> = <new value>
```

Example:

```
 UPDATE "alanparadise/nw"."employees"
  SET firstname = 'Timmy',
      titleofcourtesy = 'Mr.'
  WHERE employeeid = 20
```

SQL – Update



employeeid INTEGER	lastname VARCHAR	firstname VARCHAR	title VARCHAR	titleofcou VARCHAR
6	Suyama	Michael	Sales Representative	Mr.
7	King	Robert	Sales Representative	Mr.
8	Callahan	Laura	Inside Sales Coordinator	Ms.
9	Dodsworth	Anne	Sales Representative	Ms.
10	Dunn	Nat	Sales Representative	Mr.
21	Thomas	Tommy	Data Analyst	Mr.
20	Thomas	Timmy	Data Scientist	Mr.

New Values

SQL – Removing Rows

TRUNCATE statement – removes all rows, keeps structure

```
TRUNCATE TABLE <table name>
```

DROP statement -- removes all rows, removes structure

```
DROP TABLE <table name>
```

SQL – Removing Rows

DELETE statement – removes rows based on a condition

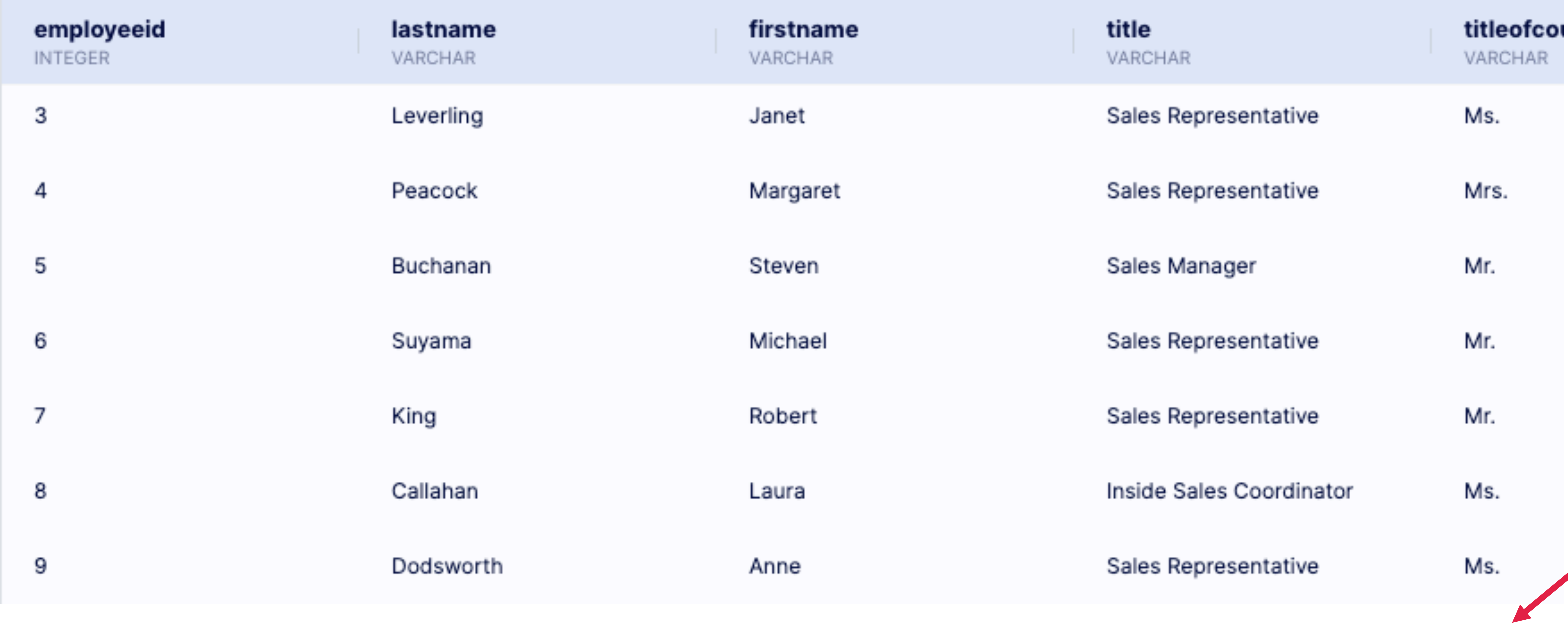
```
  DELETE FROM <table name>
      WHERE <condition>
```

Example:

```
 DELETE FROM "alanparadise/nw"."employees"
  WHERE employeeid in (10, 20, 21)
```

NOTE: a DELETE without a FROM is the same as a TRUNCATE

SQL – Removing Rows



employeeid	lastname	firstname	title	titleofco
INTEGER	VARCHAR	VARCHAR	VARCHAR	VARCHAR
3	Leverling	Janet	Sales Representative	Ms.
4	Peacock	Margaret	Sales Representative	Mrs.
5	Buchanan	Steven	Sales Manager	Mr.
6	Suyama	Michael	Sales Representative	Mr.
7	King	Robert	Sales Representative	Mr.
8	Callahan	Laura	Inside Sales Coordinator	Ms.
9	Dodsworth	Anne	Sales Representative	Ms.

New Rows are Gone

SQL – Bulk Insert

Bulk Insert (or "load") can populate a table with many rows of data.

First, let's create a new table.

```
CREATE TABLE "alanparadise/nw"."items"
  (
    itemID     INT         NOT NULL ,
    itemcode   VARCHAR(5)  NULL,
    itemname   VARCHAR(40) NOT NULL DEFAULT ' ',
    quantity   INT         NOT NULL DEFAULT 0,
    price      REAL     NOT NULL DEFAULT 0
  );
```

SQL – Bulk Insert

And then fill it with data using a subquery

NOTE: the answer set from the subquery must exactly match the table definition (column data_type and length)

```
INSERT INTO "alanparadise/nw"."items"
  (SELECT productid, concat(supplierid, categoryid, discontinued),
          productname, unitsinstock, unitprice
    from "alanparadise/nw"."products")
```

Then Verify it:

```
SELECT * FROM "alanparadise/nw"."items"
```