

A modern conference room with large windows and a blue tint. The room features a long table, several chairs, and a laptop on the table. The text "Data Manipulation Language (DML)" is overlaid in the center.

Data Manipulation Language (DML)

Data Manipulation Language

- Data Manipulation Language (DML)

its statements allow us to manipulate the data in the tables of a database

- the SELECT statement

used to retrieve data from database objects, like tables

Data Manipulation Language



SQL

```
SELECT * FROM sales;
```

sales

purchase_number

Data Manipulation Language

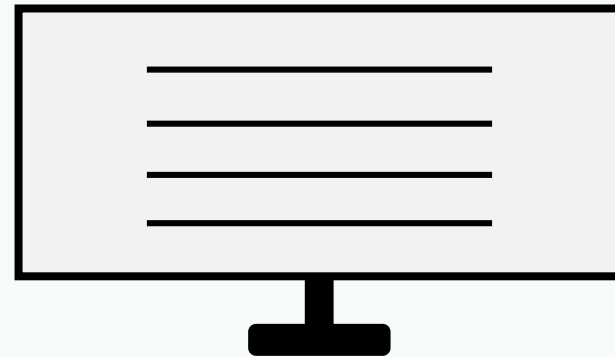


SQL

```
SELECT * FROM sales;
```

sales

purchase_number



Data Manipulation Language



SQL

```
SELECT... FROM sales...;
```

sales

purchase_number

Data Manipulation Language



SQL

```
SELECT... FROM sales...;
```

sales

purchase_number

Data Manipulation Language

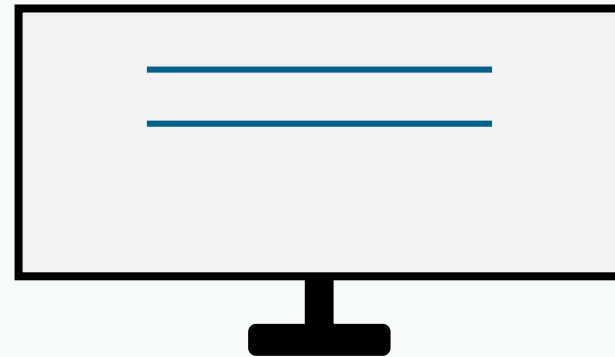


SQL

```
SELECT... FROM sales...;
```

sales

purchase_number



Data Manipulation Language

- Why are we going to need just a piece of the table?
 - imagine a table with 2 million rows of data
 - it can be helpful if you could extract only a portion of the table that satisfies given criteria
 - you should know how to use SELECT perfectly well

Data Manipulation Language

- the INSERT statement
used to insert data into tables
- INSERT... INTO... VALUES...;

Data Manipulation Language



SQL

```
INSERT INTO sales (purchase_number, date_of_purchase) VALUES  
(1, '2017-10-11');
```

sales

purchase_number	date_of_purchase

Data Manipulation Language



SQL

```
INSERT INTO sales (purchase_number, date_of_purchase) VALUES  
(1, '2017-10-11');
```

sales

purchase_number	date_of_purchase
1	2017-10-11

Data Manipulation Language



SQL

```
INSERT INTO sales VALUES  
(1, '2017-10-11');
```

sales

purchase_number	date_of_purchase
1	2017-10-11

Data Manipulation Language



SQL

```
INSERT INTO sales (purchase_number, date_of_purchase) VALUES  
(1, '2017-10-11');
```

```
INSERT INTO sales VALUES  
(1, '2017-10-11');
```

Data Manipulation Language



SQL

```
INSERT INTO sales (purchase_number, date_of_purchase) VALUES  
(2, '2017-10-27');
```

sales

purchase_number	date_of_purchase
1	2017-10-11
2	2017-10-27

Data Manipulation Language

- the UPDATE statement

allows you to renew existing data of your tables

Data Manipulation Language



SQL

sales

purchase_number	date_of_purchase
1	2017-10-11
2	2017-10-27

Data Manipulation Language



SQL

```
UPDATE sales  
SET date_of_purchase = '2017-12-12'  
WHERE purchase_number = 1;
```

sales

purchase_number	date_of_purchase
1	2017-10-11
2	2017-10-27

Data Manipulation Language



SQL

```
UPDATE sales  
SET date_of_purchase = '2017-12-12'  
WHERE purchase_number = 1;
```

sales

purchase_number	date_of_purchase
1	2017-12-12
2	2017-10-27

Data Manipulation Language

- the DELETE statement

- functions similarly to the TRUNCATE statement

- TRUNCATE vs. DELETE

TRUNCATE allows us to remove all the records contained in a table

vs.

with DELETE, you can specify precisely what you would like to be removed

Data Manipulation Language



SQL

```
DELETE FROM sales;
```

sales

purchase_number	date_of_purchase
1	2017-10-11
2	2017-10-27

Data Manipulation Language



SQL

```
DELETE FROM sales;
```

```
TRUNCATE TABLE sales;
```

sales

purchase_number	date_of_purchase
1	2017-10-11
2	2017-10-27

Data Manipulation Language



SQL

```
DELETE FROM sales;
```

```
TRUNCATE TABLE sales;
```

sales

purchase_number	date_of_purchase
1	2017-10-11
2	2017-10-27

Data Manipulation Language



SQL

```
DELETE FROM sales
WHERE
    purchase_number = 1;
```

sales

purchase_number	date_of_purchase
1	2017-10-11
2	2017-10-27

Data Manipulation Language



SQL

```
DELETE FROM sales
WHERE
    purchase_number = 1;
```

sales

purchase_number	date_of_purchase
1	2017-10-11
2	2017-10-27

Data Manipulation Language

Data Manipulation Language (DML)

- SELECT... FROM...
- INSERT INTO... VALUES
- UPDATE... SET... WHERE
- DELETE... FROM... WHERE

Next:

Data Control Language (DCL)