

SQL

Theory

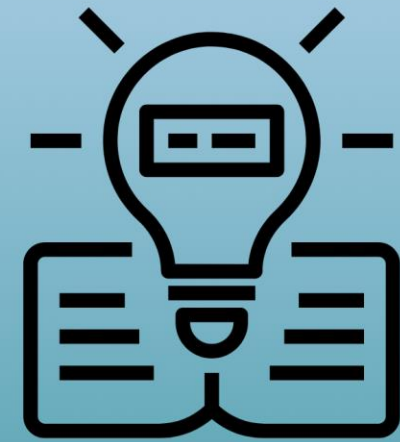


TABLE OF CONTENTS

LET'S BREAK THINGS DOWN

1. SQL THEORY

- Data Definition Language (DDL)
- SQL Keywords
- Data Manipulation Language (DML)
- Data Control Language (DCL)
- Transaction Control Language (TCL)
- SQL Syntax

DATA DEFINITION LANGUAGE (DDL)

Data Definition Language

SQL's syntax

comprises several types of statements that allow you to perform various commands and operations

Data Definition Language (DDL)

- a syntax
- a set of statements that allow the user to define or modify data structures and objects, such as tables

the CREATE statement

used for creating entire databases and database objects as tables

Data Definition Language

the CREATE statement

used for creating entire databases and database objects as tables



SQL

```
CREATE object_type object_name;
```

Data Definition Language

the CREATE statement

used for creating entire databases and database objects as tables



```
CREATE object_type object_name;
```

SQL

```
CREATE TABLE object_name (column_name data_type);
```

Data Definition Language



SQL

```
CREATE TABLE object_name (column_name data_type) ;
```

Data Definition Language



SQL

```
CREATE TABLE object_name (column_name data_type);
```

```
CREATE TABLE sales (purchase_number INT);
```


Data Definition Language



SQL

```
CREATE TABLE object_name (column_name data_type);
```

```
CREATE TABLE sales (purchase_number INT);
```

sales

purchase_number

Data Definition Language



SQL

```
CREATE TABLE sales (purchase_number INT);
```

sales

purchase_number

the table name can coincide with the name assigned to the database

Data Definition Language

the ALTER statement

used when altering existing objects

- ADD
- REMOVE
- RENAME

Data Definition Language



SQL

```
ALTER TABLE sales
```

```
ADD COLUMN date_of_purchase DATE;
```

sales

purchase_number

Data Definition Language



SQL

```
ALTER TABLE sales
```

```
ADD COLUMN date_of_purchase DATE;
```

sales

<code>purchase_number</code>	<code>date_of_purchase</code>

Data Definition Language

the DROP statement

used for deleting a database object

Data Definition Language



SQL

```
DROP object_type object_name;
```

customers

customer_id	first_name

Data Definition Language

used for deleting a database object



SQL

```
DROP object_type object_name;
```

```
DROP TABLE customers;
```

customers

customer_id	first_name

Data Definition Language

used for deleting a database object



SQL

```
DROP object_type object_name;
```

```
DROP TABLE customers;
```

customers

customer_id	first_name

Data Definition Language

the RENAME statement

allows you to rename an object

Data Definition Language



SQL

```
RENAME object_type object_name TO new_object_name;
```

customers

customer_id	first_name

Data Definition Language

used for deleting a database object



SQL

```
RENAME object_type object_name TO new_object_name;
```

```
RENAME TABLE customers TO customer_data;
```

customers

customer_id	first_name

Data Definition Language

used for deleting a database object



SQL

```
RENAME object_type object_name TO new_object_name;
```

```
RENAME TABLE customers TO customer_data;
```

customer_id	first_name

Data Definition Language

used for deleting a database object



SQL

```
RENAME object_type object_name TO new_object_name;
```

```
RENAME TABLE customers TO customer_data;
```

customer_data

customer_id	first_name

Data Definition Language

the TRUNCATE statement

instead of deleting an entire table through DROP, we can also remove its data and continue to have the table as an object in the database

Data Definition Language



SQL

```
TRUNCATE object_type object_name;
```

customers

customer_id	first_name
_____	_____
_____	_____
_____	_____
_____	_____

Data Definition Language

used for deleting a database object



SQL

```
TRUNCATE object_type object_name;
```

```
TRUNCATE TABLE customers;
```

customers

customer_id	first_name
_____	_____
_____	_____
_____	_____
_____	_____

Data Definition Language

used for deleting a database object



SQL

```
TRUNCATE object_type object_name;
```

```
TRUNCATE TABLE customers;
```

customers

customer_id	first_name
_____	_____
_____	_____
_____	_____
_____	_____

Data Definition Language

Data Definition Language (DDL)

- CREATE
- ALTER
- DROP
- RENAME
- TRUNCATE

SQL Keywords



Keywords

Keywords:

- ADD
- CREATE
- ALTER
- etc.

KEYWORDS IN SQL CANNOT BE VARIABLE NAMES!

objects or databases cannot have names that coincide with SQL keywords

Keywords

CREATE, ALTER:

Keywords

CREATE, ALTER:



SQL

```
CREATE TABLE alter (purchase_number INT) ;
```

alter

purchase_number

Data Definition Language

ADD

Data Definition Language

ADD



SQL

```
ALTER TABLE sales
```

```
ADD COLUMN date_of_purchase DATE;
```

sales

purchase_number	date_of_purchase

Data Definition Language

ADD, ALTER



SQL

```
ALTER TABLE sales
```

```
ADD COLUMN date_of_purchase DATE;
```

sales

purchase_number	date_of_purchase

Keywords

Keywords = reserved words

they cannot be used when naming objects

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

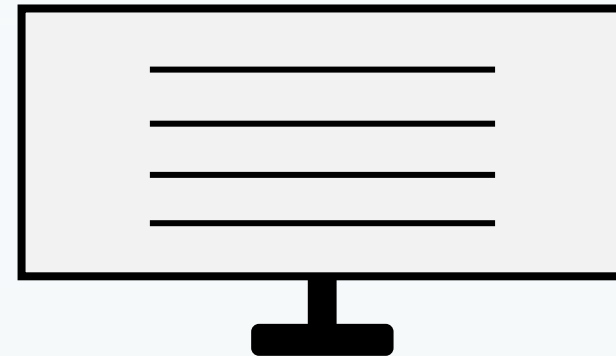


SQL

```
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

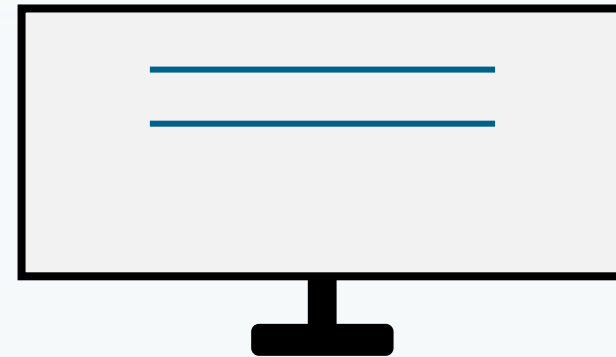
Data Manipulation Language



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

<code>purchase_number</code>	<code>date_of_purchase</code>
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...

DATA CONTROL LANGUAGE (DCL)

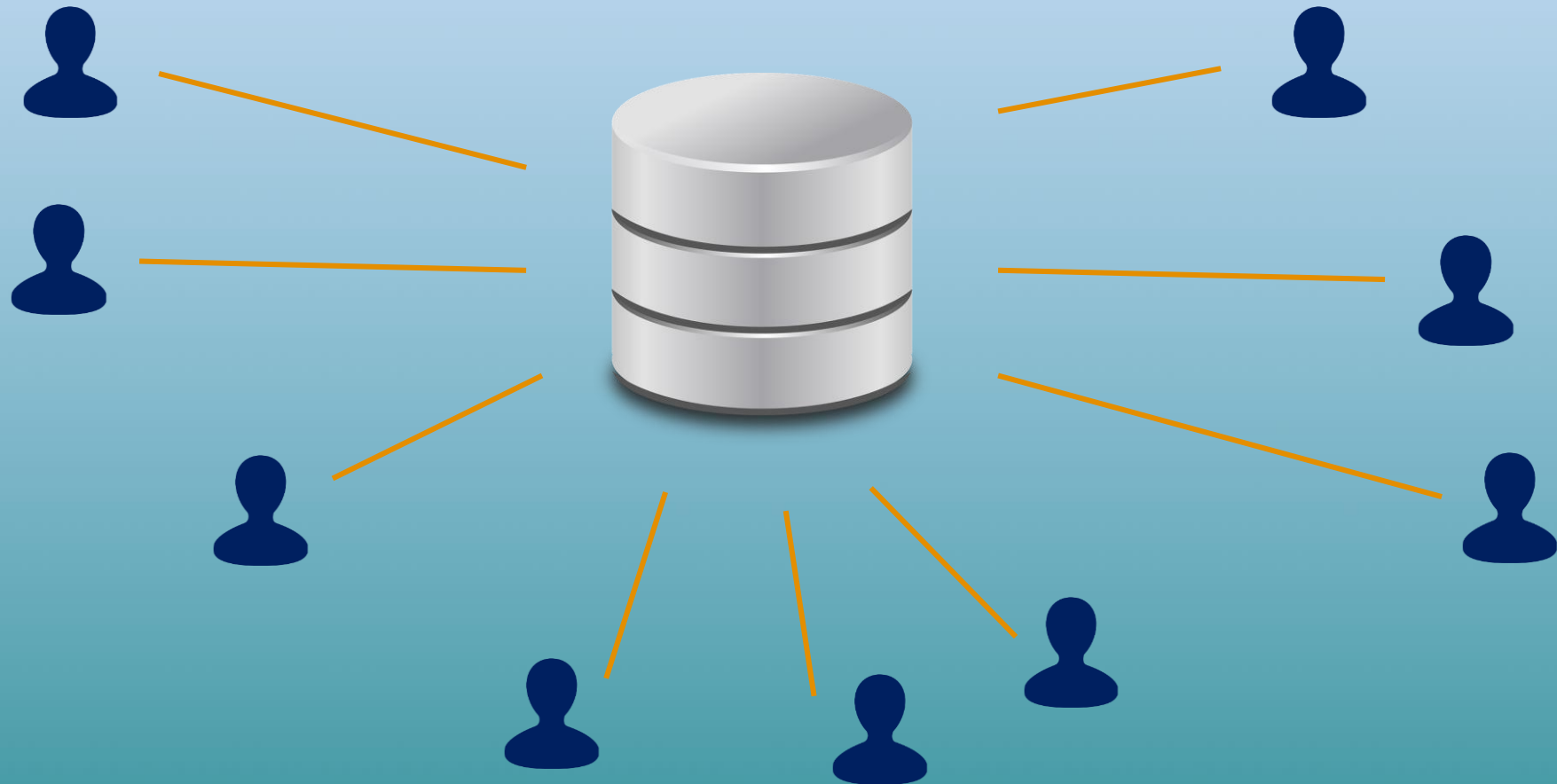
Data Control Language

Data Control Language (DCL)

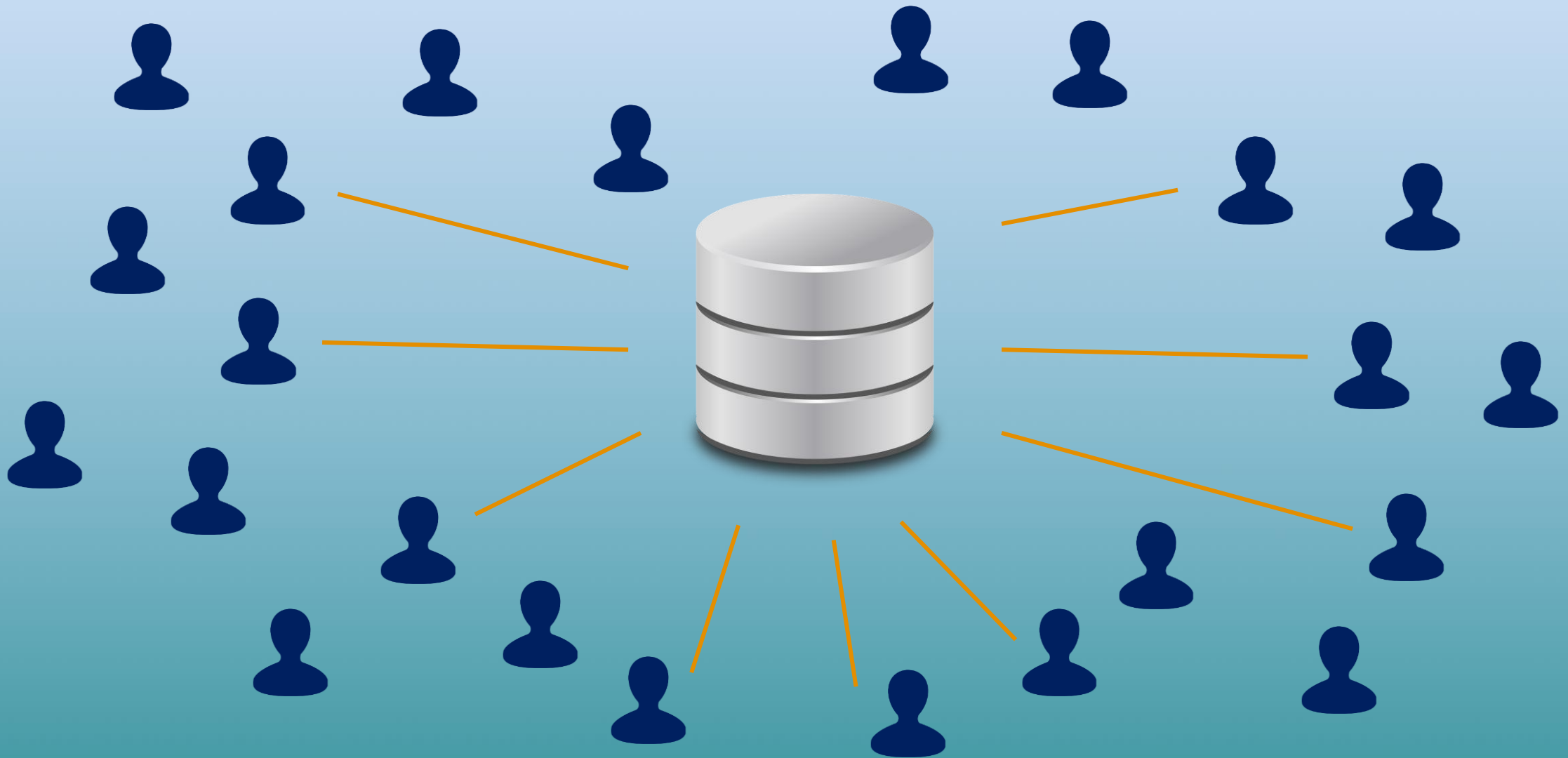
the GRANT and REVOKE statements

allow us to manage the rights users have in a database

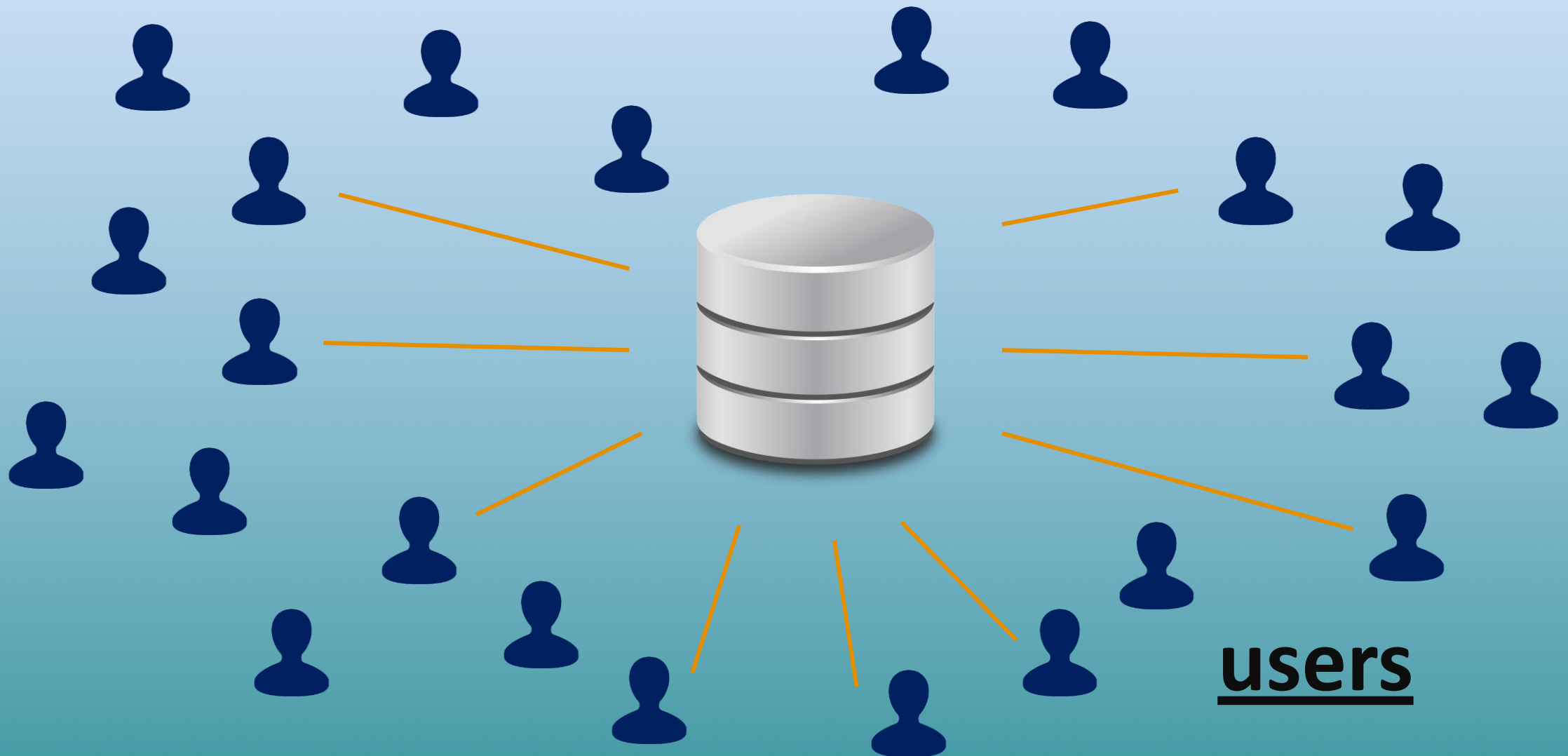
Data Control Language



Data Control Language



Data Control Language



Data Control Language

The GRANT statement

gives (or grants) certain permissions to users

Data Control Language

The GRANT statement

gives (or grants) certain permissions to users



SQL

Data Control Language

The GRANT statement

gives (or grants) certain permissions to users



SQL

```
GRANT type_of_permission ON database_name.table_name TO  
'username'@'localhost'
```

Data Control Language

The GRANT statement

gives (or grants) certain permissions to users

one can grant a *specific* type of permission, like *complete* or *partial access*



SQL

```
GRANT type_of_permission ON database_name.table_name TO  
'username'@'localhost'
```

Data Control Language

these rights will be assigned to a person who has a *username* registered at the *local server* (*'localhost': IP 127.0.0.1*)

big companies and corporations don't use this type of server, and their databases lay on *external*, more powerful servers



SQL

```
GRANT type_of_permission ON database_name.table_name TO  
'username'@'localhost'
```

Data Control Language

Database administrators

people who have *complete* rights to a database

- they can grant access to users and can revoke it

the REVOKE clause

used to revoke permissions and privileges of database users

- the exact opposite of GRANT

Data Control Language

the REVOKE clause

used to revoke permissions and privileges of database users



SQL

Data Control Language

the REVOKE clause

used to revoke permissions and privileges of database users



SQL

```
REVOKE type_of_permission ON database_name.table_name FROM  
'username' @'localhost'
```

TRANSACTION CONTROL LANGUAGE (TCL)

Transaction Control Language

Transaction Control Language (TCL)

- not every change you make to a database is saved automatically

the COMMIT statement

- related to INSERT, DELETE, UPDATE
- will save the changes you've made
- will let other users have access to the modified version of the database

Transaction Control Language

DB administrator

Customers				
customer_id	first_name	last_name	email_address	number_of_complaints
1	John	McKinley	john.mackinley@365careers.com	0
2	Elizabeth	McFarlane	e.mcfarlane@365careers.com	2
3	Kevin	Lawrence	kevin.lawrence@365careers.com	1
4	Catherine	Winnfield	c.winnfield@365careers.com	0

Transaction Control Language

DB administrator

- Change the last name of the 4th customer from 'Winnfield' to 'Johnson'

Customers				
customer_id	first_name	last_name	email_address	number_of_complaints
1	John	McKinley	john.mackinley@365careers.com	0
2	Elizabeth	McFarlane	e.mcfarlane@365careers.com	2
3	Kevin	Lawrence	kevin.lawrence@365careers.com	1
4	Catherine	Winnfield	c.winnfield@365careers.com	0

Transaction Control Language

DB administrator

- Change the last name of the 4th customer from 'Winnfield' to 'Johnson'

Customers				
customer_id	first_name	last_name	email_address	number_of_complaints
1	John	McKinley	john.mackinley@365careers.com	0
2	Elizabeth	McFarlane	e.mcfarlane@365careers.com	2
3	Kevin	Lawrence	kevin.lawrence@365careers.com	1
4	Catherine		c.winnfield@365careers.com	0

Transaction Control Language

DB administrator

- Change the last name of the 4th customer from 'Winnfield' to 'Johnson'

Customers				
customer_id	first_name	last_name	email_address	number_of_complaints
1	John	McKinley	john.mackinley@365careers.com	0
2	Elizabeth	McFarlane	e.mcfarlane@365careers.com	2
3	Kevin	Lawrence	kevin.lawrence@365careers.com	1
4	Catherine	Johnson	c.winnfield@365careers.com	0

Transaction Control Language

DB administrator



SQL

Customers					
customer_id	first_name	last_name	email_address	number_of_complaints	
1	John	McKinley	john.mackinley@365careers.com	0	
2	Elizabeth	McFarlane	e.mcfarlane@365careers.com	2	
3	Kevin	Lawrence	kevin.lawrence@365careers.com	1	
4	Catherine	Winnfield	c.winnfield@365careers.com	0	

Transaction Control Language

DB administrator



SQL

```
UPDATE customers
SET last_name = 'Johnson'
WHERE customer_id = 4;
```

Customers					
customer_id	first_name	last_name	email_address	number_of_complaints	
1	John	McKinley	john.mackinley@365careers.com	0	
2	Elizabeth	McFarlane	e.mcfarlane@365careers.com	2	
3	Kevin	Lawrence	kevin.lawrence@365careers.com	1	
4	Catherine	Winnfield	c.winnfield@365careers.com	0	

Transaction Control Language

DB administrator



SQL

```
UPDATE customers
SET last_name = 'Johnson'
WHERE customer_id = 4;
```

Customers					
customer_id	first_name	last_name	email_address	number_of_complaints	
1	John	McKinley	john.mackinley@365careers.com	0	
2	Elizabeth	McFarlane	e.mcfarlane@365careers.com	2	
3	Kevin	Lawrence	kevin.lawrence@365careers.com	1	
4	Catherine		c.winnfield@365careers.com	0	

Transaction Control Language

DB administrator



SQL

```
UPDATE customers
SET last_name = 'Johnson'
WHERE customer_id = 4;
```

Customers				
customer_id	first_name	last_name	email_address	number_of_complaints
1	John	McKinley	john.mackinley@365careers.com	0
2	Elizabeth	McFarlane	e.mcfarlane@365careers.com	2
3	Kevin	Lawrence	kevin.lawrence@365careers.com	1
4	Catherine	Johnson	c.winnfield@365careers.com	0

Transaction Control Language

DB administrator

Customers				
customer_id	first_name	last_name	email_address	number_of_complaints
1	John	McKinley	john.mackinley@365careers.com	0
2	Elizabeth	McFarlane	e.mcfarlane@365careers.com	2
3	Kevin	Lawrence	kevin.lawrence@365careers.com	1
4	Catherine	Johnson	c.winnfield@365careers.com	0

Problem:

users

Customers				
customer_id	first_name	last_name	email_address	number_of_complaints
1	John	McKinley	john.mackinley@365careers.com	0
2	Elizabeth	McFarlane	e.mcfarlane@365careers.com	2
3	Kevin	Lawrence	kevin.lawrence@365careers.com	1
4	Catherine	Winnfield	c.winnfield@365careers.com	0

Transaction Control Language

DB administrator



SQL

```
UPDATE customers
SET last_name = 'Johnson'
WHERE customer_id = 4;
```

Customers				
customer_id	first_name	last_name	email_address	number_of_complaints
1	John	McKinley	john.mackinley@365careers.com	0
2	Elizabeth	McFarlane	e.mcfarlane@365careers.com	2
3	Kevin	Lawrence	kevin.lawrence@365careers.com	1
4	Catherine	Johnson	c.winnfield@365careers.com	0

Transaction Control Language

DB administrator



SQL

```
UPDATE customers
SET last_name = 'Johnson'
WHERE customer_id = 4
COMMIT;
```

Customers					
customer_id	first_name	last_name	email_address	number_of_complaints	
1	John	McKinley	john.mackinley@365careers.com	0	
2	Elizabeth	McFarlane	e.mcfarlane@365careers.com	2	
3	Kevin	Lawrence	kevin.lawrence@365careers.com	1	
4	Catherine	Johnson	c.winnfield@365careers.com	0	

Transaction Control Language

DB administrator

Customers				
customer_id	first_name	last_name	email_address	number_of_complaints
1	John	McKinley	john.mackinley@365careers.com	0
2	Elizabeth	McFarlane	e.mcfarlane@365careers.com	2
3	Kevin	Lawrence	kevin.lawrence@365careers.com	1
4	Catherine	Johnson	c.winnfield@365careers.com	0

users

Customers				
customer_id	first_name	last_name	email_address	number_of_complaints
1	John	McKinley	john.mackinley@365careers.com	0
2	Elizabeth	McFarlane	e.mcfarlane@365careers.com	2
3	Kevin	Lawrence	kevin.lawrence@365careers.com	1
4	Catherine	Johnson	c.winnfield@365careers.com	0

Transaction Control Language

the COMMIT statement

committed states can accrue

the ROLLBACK clause

the clause that will let you make a step back

- allows you to undo any changes you have made but don't want to be saved permanently

Transaction Control Language

DB administrator



SQL

```
UPDATE customers
SET last_name = 'Johnson'
WHERE customer_id = 4
COMMIT;
```

Customers					
customer_id	first_name	last_name	email_address	number_of_complaints	
1	John	McKinley	john.mackinley@365careers.com	0	
2	Elizabeth	McFarlane	e.mcfarlane@365careers.com	2	
3	Kevin	Lawrence	kevin.lawrence@365careers.com	1	
4	Catherine	Johnson	c.winnfield@365careers.com	0	

Transaction Control Language

DB administrator



SQL

```
UPDATE customers
SET last_name = 'Johnson'
WHERE customer_id = 4
COMMIT;

ROLLBACK;
```

Customers					
customer_id	first_name	last_name	email_address	number_of_complaints	
1	John	McKinley	john.mackinley@365careers.com	0	
2	Elizabeth	McFarlane	e.mcfarlane@365careers.com	2	
3	Kevin	Lawrence	kevin.lawrence@365careers.com	1	
4	Catherine	Johnson	c.winnfield@365careers.com	0	

Transaction Control Language

DB administrator



SQL

```
UPDATE customers
SET last_name = 'Johnson'
WHERE customer_id = 4
COMMIT;

ROLLBACK;
```

Customers					
customer_id	first_name	last_name	email_address	number_of_complaints	
1	John	McKinley	john.mackinley@365careers.com	0	
2	Elizabeth	McFarlane	e.mcfarlane@365careers.com	2	
3	Kevin	Lawrence	kevin.lawrence@365careers.com	1	
4	Catherine	Winnfield	c.winnfield@365careers.com	0	

Transaction Control Language

the COMMIT statement

- saves the transaction in the database
- changes cannot be undone

the ROLLBACK clause

- allows you to take a step back
- the last change(s) made will not count
- reverts to the last non-committed state

SQL Syntax

DDL – Data Definition Language

creation of data

DML – Data Manipulation Language

manipulation of data

DCL – Data Control Language


assignment and removal of permissions to use this data

TCL – Transaction Control Language

saving and restoring changes to a database

Want to master SQL skills?

Explore our SQL courses, ranging from beginner to advanced levels. Sign up now to try the first 2 course sections for free!




SQL

with Martin Ganchev, Vladimi...

4.8/5 ★★★★★ (3250)

12 hours • 121 lessons

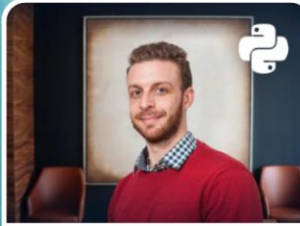


Advanced SQL

with Martin Ganchev, Vladimi...

4.8/5 ★★★★★ (320)

7 hours • 20 lessons



SQL + Tableau + Python

with Martin Ganchev

4.8/5 ★★★★★ (348)

5 hours • 61 lessons

Sign Up Now