

# Intro to SQL

Short for Structured Query Language

Pronounced "S-Q-L" or "sequel"

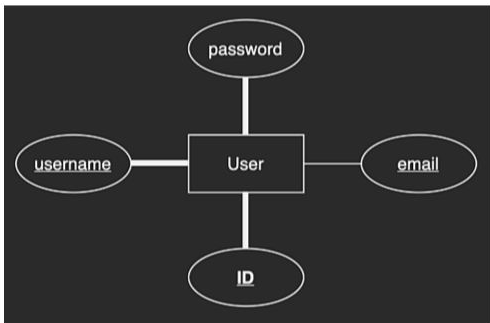
Domain Specific Language (DSL)

Primary use is to Create, Read, Update, or Delete data

Also used to construct database table architectures

# Creating a table with SQL

Start with ER diagram



Define tables  
& columns

id	username	password	email
1	alice123	1234	
2	bob45	p455w0rd	
3	carol995	1234	carol@example.com
4	dave	mysteryw0rd	dave@example.com
5	edgar	data4life	



Create with SQL

?

# SQL CREATE syntax

Basic syntax

```
CREATE TABLE table_name (  
    column1 datatype column_constraints,  
    column2 datatype column_constraints,  
    column3 datatype column_constraints,  
    PRIMARY KEY(column_names)  
);
```

## Full syntax from Postgres documentation

```
CREATE [ [ GLOBAL | LOCAL ] { TEMPORARY | TEMP } | UNLOGGED ] TABLE [ IF NOT EXISTS ] table_name ( [
    { column_name data_type [ COLLATE collation ] [ column_constraint [ ... ] ]
    | table_constraint
    | LIKE parent_table [ like_option ... ] }
    [, ... ]
] )
[ INHERITS ( parent_table [, ... ] ) ]
[ WITH ( storage_parameter [= value] [, ... ] ) | WITHOUT OIDS | WITHOUT OIDS ]
[ ON COMMIT { PRESERVE ROWS | DELETE ROWS | DROP } ]
[ TABLESPACE tablespace ]
```

# Choosing a data type

python type	postgres type
<code>int</code>	<code>INTEGER, INT</code>
sequential <code>int</code>	<code>SERIAL</code>
<code>float</code>	<code>NUMERIC</code>
<code>string</code>	<code>TEXT</code>
<code>string</code> of length X	<code>CHARACTER(X)</code>
<code>boolean</code>	<code>BOOLEAN, BOOL</code>
<code>datetime</code>	<code>TIMESTAMP</code>

Consider what it would be in Python,  
and find corresponding type

Consider special types provided by  
Postgres such as `SERIAL` or  
`CHARACTER(X)`

```
CREATE TABLE table_name (  
    column1 datatype column_constraints,  
    column2 datatype column_constraints,  
    column3 datatype column_constraints,  
    PRIMARY KEY(column_names)  
);
```

# Choosing column constraints

column constraint	SQL notation
column has different value for every row	<b>UNIQUE</b>
column cannot be blank	<b>NOT NULL</b>
default value if none is provided	<b>DEFAULT</b>
unique row identifier	<b>PRIMARY KEY</b>

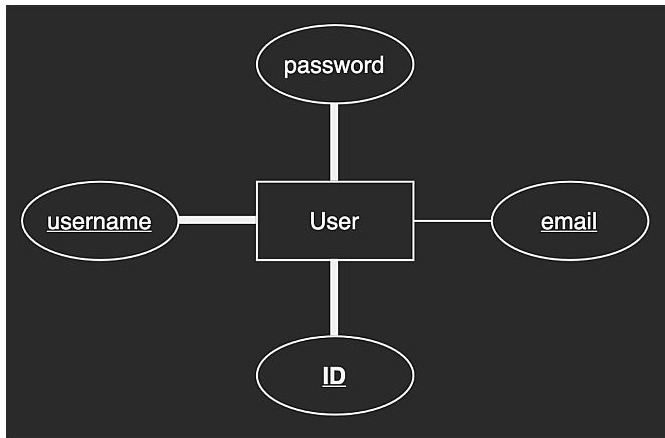
Added after data type, separated by spaces

Primary key should have its own line

Primary key column names must match other column names, separated by commas

```
CREATE TABLE table_name (  
    column1 datatype column_constraints,  
    column2 datatype column_constraints,  
    column3 datatype column_constraints,  
    PRIMARY KEY(column_names)  
);
```

# Choosing column constraints



attribute	postgres type	column_constraints
id	SERIAL	PRIMARY KEY
username	TEXT	NOT NULL, UNIQUE
password	TEXT	NOT NULL
email	TEXT	UNIQUE

```

CREATE TABLE table_name (
  column1 datatype column_constraints,
  column2 datatype column_constraints,
  column3 datatype column_constraints,
  PRIMARY KEY(column_names)
);
    
```



```

CREATE TABLE users
(
  id SERIAL,
  username TEXT NOT NULL UNIQUE,
  password TEXT NOT NULL,
  email TEXT UNIQUE,
  PRIMARY KEY(id)
);
    
```