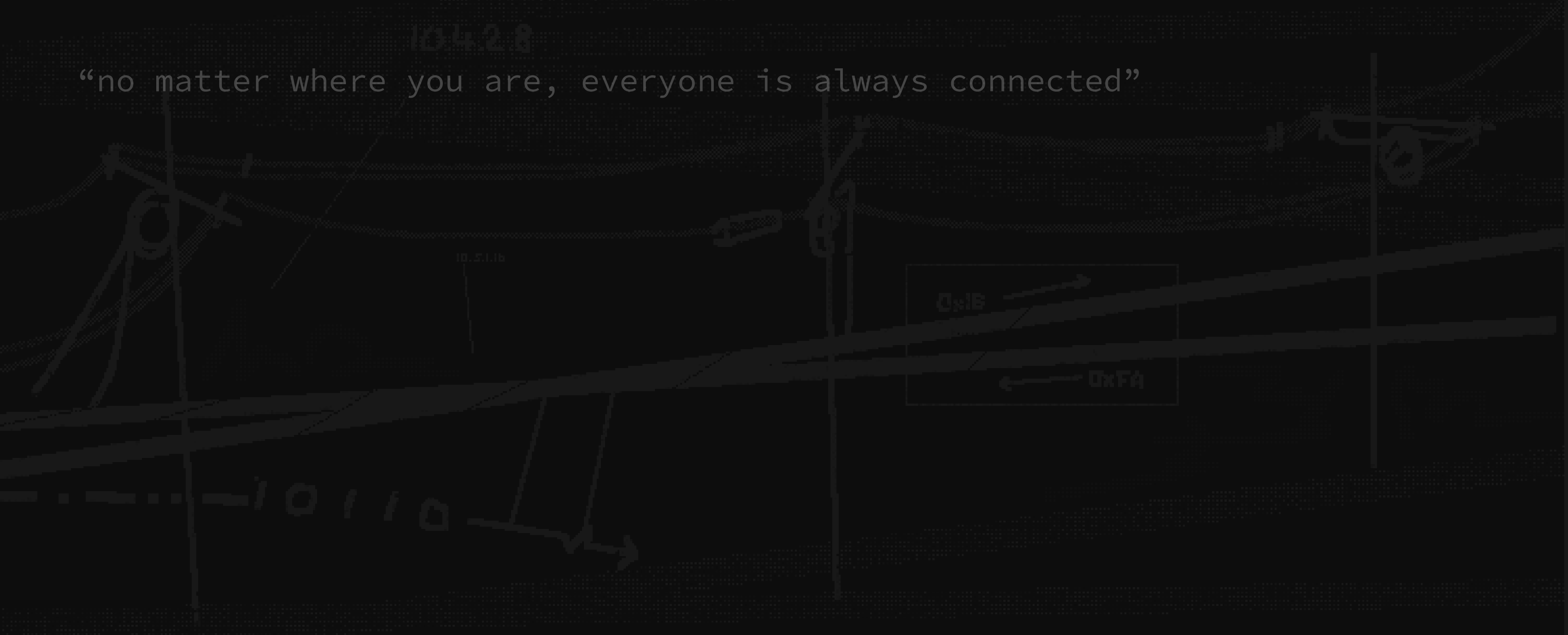


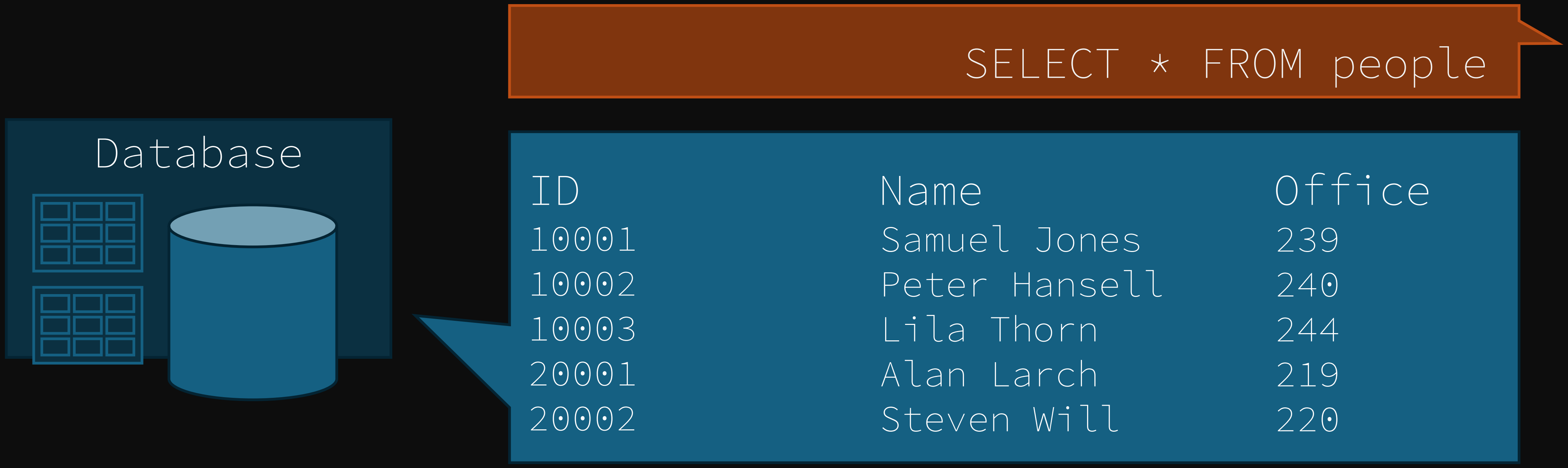
“no matter where you are, everyone is always connected”

14

MySQL Databases

What is MySQL?

MySQL is an open-source relational database management system that can be interacted with using the structured query language. Data is stored into tables like the one shown below, and tables can relate with other tables.



Database

```
SELECT * FROM people
```

ID	Name	Office
10001	Samuel Jones	239
10002	Peter Hansell	240
10003	Lila Thorn	244
20001	Alan Larch	219
20002	Steven Will	220

Relational Databases

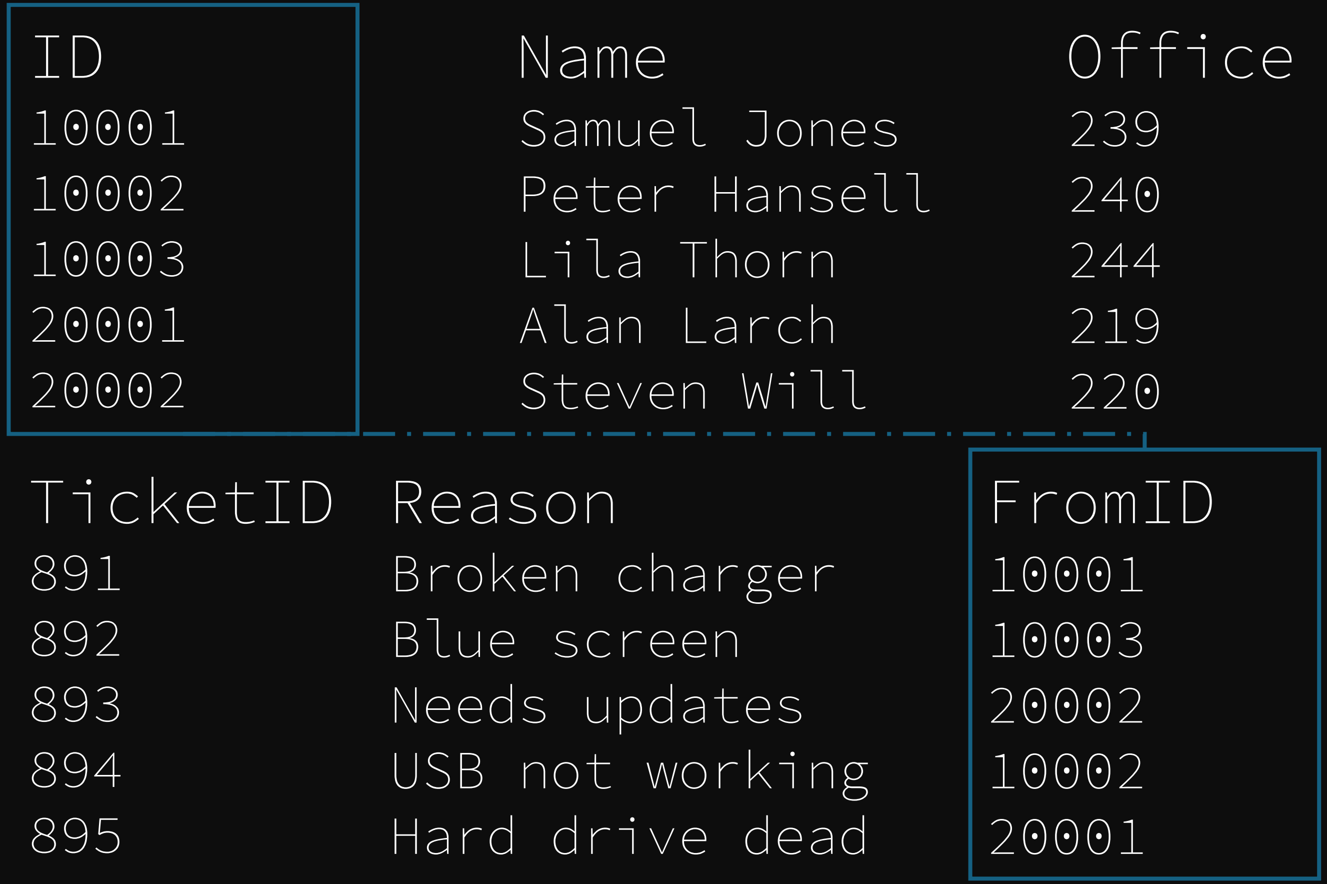
Let’s take a look at a relational database. Here, we have the table *people* shown in the previous slide.

ID	Name	Office
10001	Samuel Jones	239
10002	Peter Hansell	240
10003	Lila Thorn	244
20001	Alan Larch	219
20002	Steven Will	220

Relational Databases

Now, we have a new table *tickets*. We have a column in this new table *FromID*. Notice that these are user IDs found from the *people* table.

Because of this common thread we can interrelate entries from *tickets* to entries from *people*.



ID	Name	Office
10001	Samuel Jones	239
10002	Peter Hansell	240
10003	Lila Thorn	244
20001	Alan Larch	219
20002	Steven Will	220

TicketID	Reason	FromID
891	Broken charger	10001
892	Blue screen	10003
893	Needs updates	20002
894	USB not working	10002
895	Hard drive dead	20001

Structured Query Language

people

ID	Name	Office
10001	Samuel Jones	239
10002	Peter Hansell	240
10003	Lila Thorn	244
20001	Alan Larch	219
20002	Steven Will	220

tickets

TicketID	Reason	FromID
891	Broken charger	10001
892	Blue screen	10003
893	Needs updates	20002
894	USB not working	10002
895	Hard drive dead	20001

But how do we get and interact with this data from the database? That’s where the SQL part comes in. We can use queries to pick the data we want.

Structured Query Language

people

ID	Name	Office
10001	Samuel Jones	239
10002	Peter Hansell	240
10003	Lila Thorn	244
20001	Alan Larch	219
20002	Steven Will	220

tickets

TicketID	Reason	FromID
891	Broken charger	10001
892	Blue screen	10003
893	Needs updates	20002
894	USB not working	10002
895	Hard drive dead	20001

Let’s say, for example, we want all the names in the *people* table.

The query would be

```
SELECT Name FROM people
```

And it would return

Name
Samuel Jones
Peter Hansell
Lila Thorn
Alan Larch
Steven Will

Basic SQL Syntax

in a *mysql* command line

```
mysql> SHOW DATABASES;

+---------------------+
| Database            |
+---------------------+
| information_schema  |
| mysql               |
| office              |
| performance_schema  |
| sys                 |
+---------------------+
5 rows in set (0.00 sec)
```

Let’s continue using the previous database tables we’ve set earlier, now just in the MySQL command line.

First, let’s show all of the databases in our server. We use the SHOW DATABASES command for this. Our database is called *office*.

Basic SQL Syntax

in a *mysql* command line

```
mysql> USE office;
Database changed

mysql> show tables;
+------------------+
| Tables_in_office |
+------------------+
| people           |
| tickets          |
+------------------+
2 rows in set (0.00 sec)
```

We can enter into the database using the USE command...

... and see what tables are inside by using the SHOW TABLES command.

Basic SQL Syntax

in a *mysql* command line

```
mysql> SELECT * FROM people;
+-------+---------------+--------+
| ID    | Name          | Office |
+-------+---------------+--------+
| 10001 | Samuel Jones  |    239 |
| 10002 | Peter Hansell |    240 |
| 10003 | Lila Thorn    |    244 |
| 20001 | Alan Larch    |    219 |
| 20002 | Steven Will   |    220 |
+-------+---------------+--------+
5 rows in set (0.00 sec)
```

We can see the entirety of one table using the SHOW command and a wildcard ‘*’.

Basic SQL Syntax

in a *mysql* command line

```
mysql> SELECT Name FROM people;
+----------------+
| Name           |
+----------------+
| Samuel Jones   |
| Peter Hansell  |
| Lila Thorn     |
| Alan Larch     |
| Steven Will    |
+----------------+
5 rows in set (0.00 sec)
```

Or we can simply see the column that we want, such as *Name*.

Basic SQL Syntax

in a *mysql* command line

```
mysql> INSERT INTO people VALUES (
    -> 2003, ‘Timothy Emmanuel’, 272,
    -> );
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM people;
+-------+------------------+--------+
| ID    | Name             | Office |
+-------+------------------+--------+
| 10001 | Samuel Jones     |    239 |
                  ...
| 20003 | Timothy Emmanuel |    272 |
+-------+------------------+--------+
6 rows in set (0.00 sec)
```

We can create a new entry in this table by using the INSERT command and entering the values in order of the columns.

Basic SQL Syntax

in a *mysql* command line

```
mysql> DELETE FROM people WHERE
    -> ID=10002;
Query OK, 1 row affected (0.00 sec)

mysql> select * from people;
+-------+-------------------+--------+
| ID    | Name              | Office |
+-------+-------------------+--------+
| 10001 | Samuel Jones      |    239 |
| 10003 | Lila Thorn        |    244 |
                 ...
+-------+-------------------+--------+
5 rows in set (0.00 sec)
```

We can remove entries by using the DELETE command and specifying which rows we want to remove based on the column value.

Basic SQL Syntax

That is just a taste of what you can do with SQL. More complex tasks can be done with more complex queries.

To recap, you just learned how to...

- SHOW DATABASES
- USE [database]
- SHOW TABLES
- SELECT [columns] FROM [table]
- INSERT INTO [table] VALUES (...)
- DELETE FROM [table] WHERE [column]=[target]

Securing the Database

in terminal

```
# mysql_secure_installation
```

A good place to start is to run the built in security script for MySQL.

It’ll prompt you for a few options. Use your sense and choose the most appropriate and secure settings!

Configuration Options

in file */etc/mysql/my.cnf*

```
[Client]
  port=3306
  socket=/var/run/mysqld/mysqld.sock
```

MySQL holds it’s configuration in /etc/mysql/my.cnf.

Configure the following values.

Configuration Options

in file */etc/mysql/my.cnf*

```
[mysqld_safe]
  socket=/var/run/mysqld/mysqld.sock
  nice=0
  lc-messages-dir=/usr/share.mysql
  lc-messages=en_US
  bind-address=127.0.0.1
  key_buffer=16M
  max_allowed_packet=16M
  thread_stack=192K
  thread_cache_size=8
  myisam-recover=BACKUP
```

MySQL holds it’s configuration in /etc/mysql/my.cnf.

Configure the following values.

Configuration Options

in file */etc/mysql/my.cnf*

```
[mysqld]
  user=mysql
  pid-file=/var/run/mysqld/mysqld.pid
  socket=/var/run/mysqld/mysqld.sock
  port=3306
  basedir=/usr
  datadir=/var/lib.mysql
  tmpdir=/tmp
  query_cache_limit=1M
  query_cache_size=16M
  log_error=/var/log/mysql/error.log
  expire_log_days=10
  max_binlog_size=100M
```

MySQL holds it’s configuration in /etc/mysql/my.cnf.

Configure the following values.

Configuration Options

in file */etc/mysql/my.cnf*

```
[mysqldump]
  quick
  quote-names
  max_allowed_packet=16M
[isamchk]
  key_buffer=16M
  !includedir /etc/mysql/conf.d
```

MySQL holds it’s configuration in /etc/mysql/my.cnf.

Configure the following values.

A Note on Other Databases

Other SQL-like databases use similar syntax but differ slightly. The syntax provided here can apply to other databases, but it’s always a good idea to check the specific documentation!