Overview of SQL:

SQL itself
- LANGUAGE
- Widely used by database management SYSTEMS
- like mysql, postgresql, microsoft sql server, oracle
- for managing relational databases

Relational Databases
- Data organized in tables
- Columns represent attributes / types of data
- Each row represents an individual record or datapoint

Cont.
- Each row also has its own unique primary key
- By taking the unique id from a row and storing it in a different row in a different table
- Storing it in a special column known as a foreign key
- Purpose: form relationships between tables
- Teams table: team id is primary key
- Players table: foreign key
- Player belongs to 1 team
- Team can have many players

Non-relational Databases
- Like Redis, mongoDB
- Aka NoSQL databases
- Typically utilize their own language or commands
- Consist of any data formatted not as a table
- JSON file, strings, hashes, lists, etc.

Role of SQL
- Not only create read update delete data
- Join data from different tables together
- Forming complex relationships
- Allows you to store data in its smallest form / normalized form
- Eliminate redundancy and duplication

DEMO:

```
CREATE TABLE Users(
   id INT AUTO_INCREMENT,
   email VARCHAR(255) NOT NULL UNIQUE,
   bio TEXT,
   country VARCHAR(2)
PRIMARY KEY(id)
);
```

```
INSERT INTO users (email, bio, country)
VALUES
        ('helloworld@gmail.com', 'I love strangers!', 'US'),
   ('hola@munda.com', 'bar', 'MX'),
   ('bonjour@monde.com', 'baz', 'FR'),
   ('hello@world.com', 'foo', 'US');
```

```
CREATE INDEX email_index ON users(email);
```

```
CREATE TABLE rooms(
         INidT PRIMARY KEY AUTO_INCREMENT,
   owner_id INT,
        price INT,
   street VARCHAR(255),
   FOREIGN KEY(owner_id) REFERENCES users(id)
);
```

```
INSERT INTO rooms (owner_id, price, street)
VALUES
        (1, 200, 'san diego sailboat'),
   (1, 150, 'nantucket cottage'),
   (1, 300, 'vail cabin'),
   (1, 450, 'sf cardboard box');
```

```
SELECT
        users.id AS user_id,
   rooms.id AS room_id,
   email,
   bio,
   street
FROM users
LEFT JOIN rooms
ON rooms.owner_id = Users.id
ORDER BY user_id;
```

```
CREATE TABLE bookings(
        id INT PRIMARY KEY AUTO_INCREMENT,
   guest_id INT NOT NULL,
   room_id INT NOT NULL,
   check_in DATETIME,
   FOREIGN KEY (guest_id) REFERENCES users(id),
   FOREIGN KEY (room_id) REFERENCES rooms(id)
);
```

```
INSERT INTO bookings (guest_id, room_id, check_in)
VALUES
(2, 1, '2024-01-01'),
(3, 2, '2024-01-02'),
(2, 3, '2024-01-03'),
(2, 4, '2024-01-04');
(3, 3, ‘2024-01-04’);
```

```
-- info for all bookings by a guest
SELECT
   guest_id,
   street,
   check_in
FROM bookings
INNER JOIN rooms ON rooms.id = bookings.room_id
WHERE guest_id = 2;
```

```
-- display info on all guests who have lived in a room

SELECT
        guest_id,
    room_id,
    check_in,
    email
FROM bookings
INNER JOIN users ON users.id = guest_id
WHERE room_id = 3;
```