

# Introduction to Object-Relational Mapping

## (ORM)

A technique for performing SQL queries from a backend,  
e.g. Python, Java, Node

**Object:** Grouping of related data fields and methods e.g. Python class

**Relational:** Refers to Relational Database e.g. Postgres

**Mapping:** Translation from one domain to another, e.g. Python dictionary

**Object-relational mapping:** Translates object-oriented code into SQL

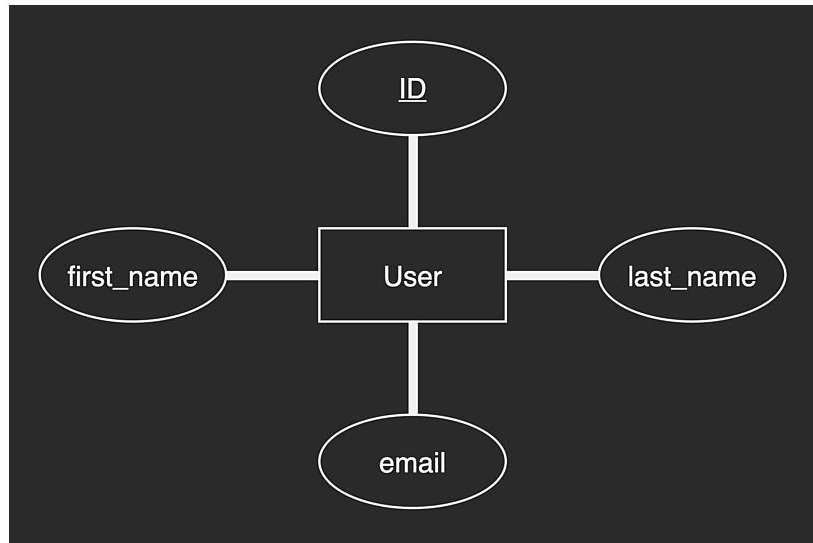
## Example

### Prompt:

Query the **full name** and **email** of the  
User with an ID of 5

### Example output:

John Doe john@example.com



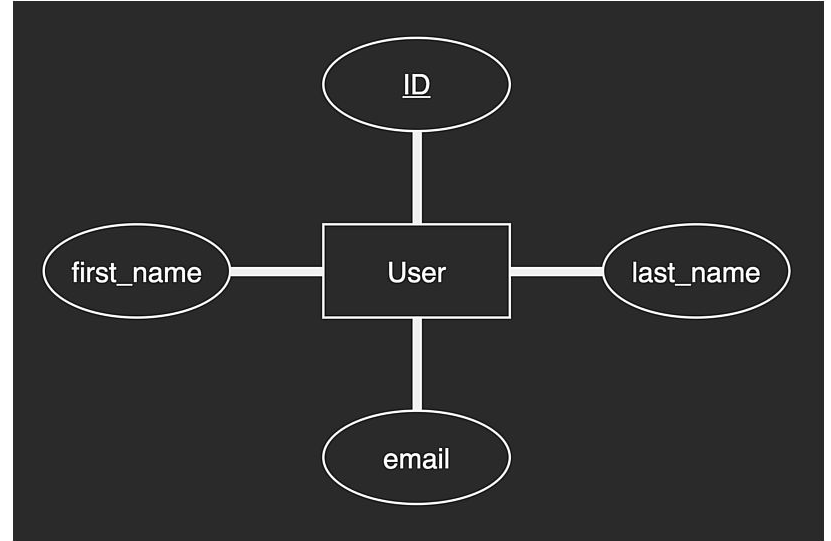
# Example

```
SELECT first_name, last_name, email  
FROM users  
WHERE id = 5;
```

Q: How to execute from backend?

A: Use **database adapter**

a.k.a. **database driver, database connector**



```
# print full name and email of user #5
```

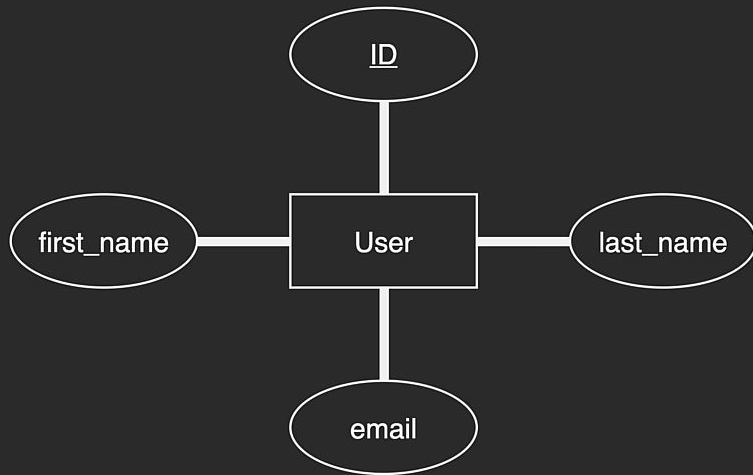
```
db = get_database() # assume this returns a database adapter  
sql = "SELECT id, first_name, last_name FROM users WHERE id = 5";  
result = db.execute(sql)
```

```
users = list(result)  
user5 = users[0]
```

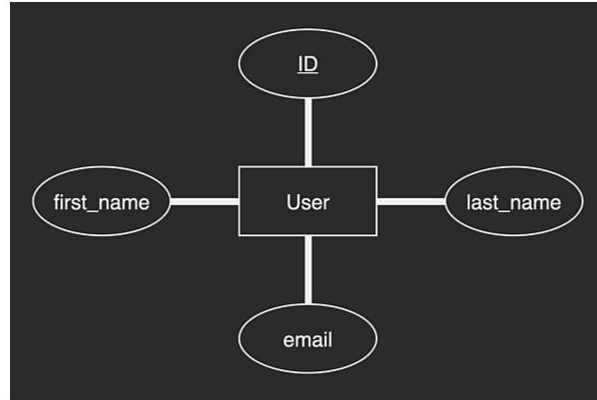
```
first_name = user5["first_name"]  
last_name = user5["last_name"]
```

```
full_name = first_name + " " + last_name  
email = user5["email"]
```

```
print(full_name, email)
```



# ORM syntax approach



```
# print full name and email of user #5

db = get_database() # assume this returns a database adapter

user5 = db.Users.get(5) # user5 is a User object

full_name = user5.get_full_name() # get_full_name is a method on the User object
email = user5.email # email is a field on the User object

print(full_name, email)
```

## Raw SQL

```
# print full name and email of user #5

db = get_database() # assume this returns a database adapter
sql = "SELECT id, first_name, last_name FROM users WHERE id = 5";
result = db.execute(sql)

users = list(result)
user5 = users[0]

first_name = user5["first_name"]
last_name = user5["last_name"]

full_name = first_name + " " + last_name
email = user5["email"]

print(full_name, email)
```

## ORM

```
# print full name and email of user #5

db = get_database() # assume this returns a database adapter

user5 = db.Users.get(5) # user5 is a User object

full_name = user5.get_full_name() # get_full_name is a method on the User object
email = user5.email # email is a field on the User object

print(full_name, email)
```