Web Programming

SQL, Models and Migrations

Prof. Josué Obregón
Department of Industrial Engineering- ITM
Seoul National University of Science and Technology

Objectives

- Identify and describe the basic SQL operations that are used for manipulating relational databases
- Define what is Django ORM and how is used to abstract the data manipulation while developing a web application
- Apply the acquired knowledge of the Django ORM to construct a dynamic web applications.

Agenda

- SQL basics
- Django ORM
  - Models and Migrations
  - Relationships
  - Queries

  - Django shell

Data

origin	destination	duration
New York	London	415
Shanghai	Paris	760
Istanbul	Tokyo	700
New York	Paris	435
Moscow	Paris	245
Lima	New York	455

SQL

Database Management Systems

- MySQL
- PostgreSQL
- SQLite
- ...

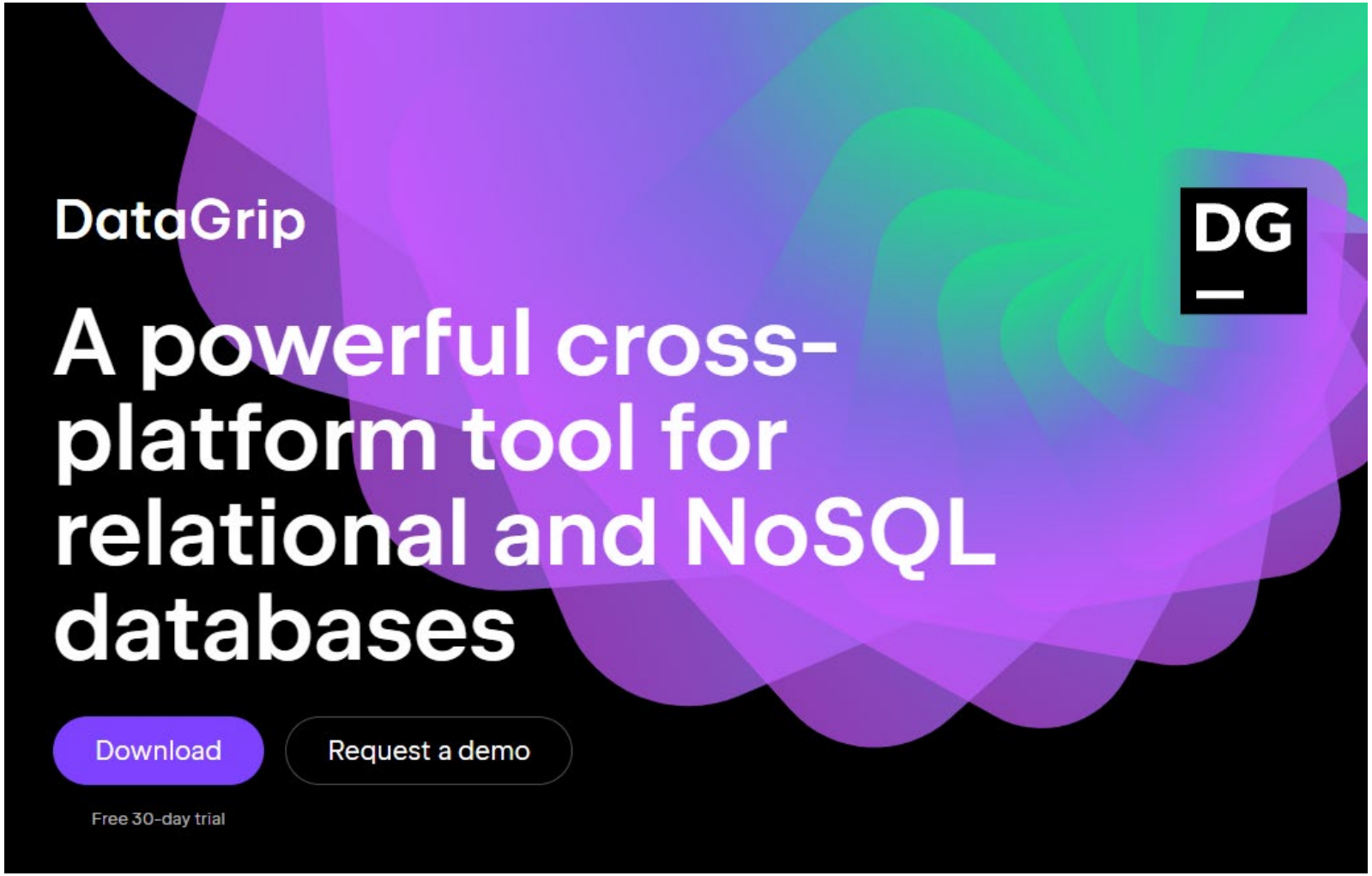
SQLite Types

- **NULL**. The value is a NULL value.
- **INTEGER**. The value is a signed integer, stored in 0, 1, 2, 3, 4, 6, or 8 bytes depending on the magnitude of the value.
- **REAL**. The value is a floating-point value, stored as an 8-byte IEEE floating point number.
- **TEXT**. The value is a text string, stored using the database encoding (UTF-8, UTF-16BE or UTF-16LE).
- **BLOB**. The value is a blob of data, stored exactly as it was input (Binary Large Object, e.g., audio and video)

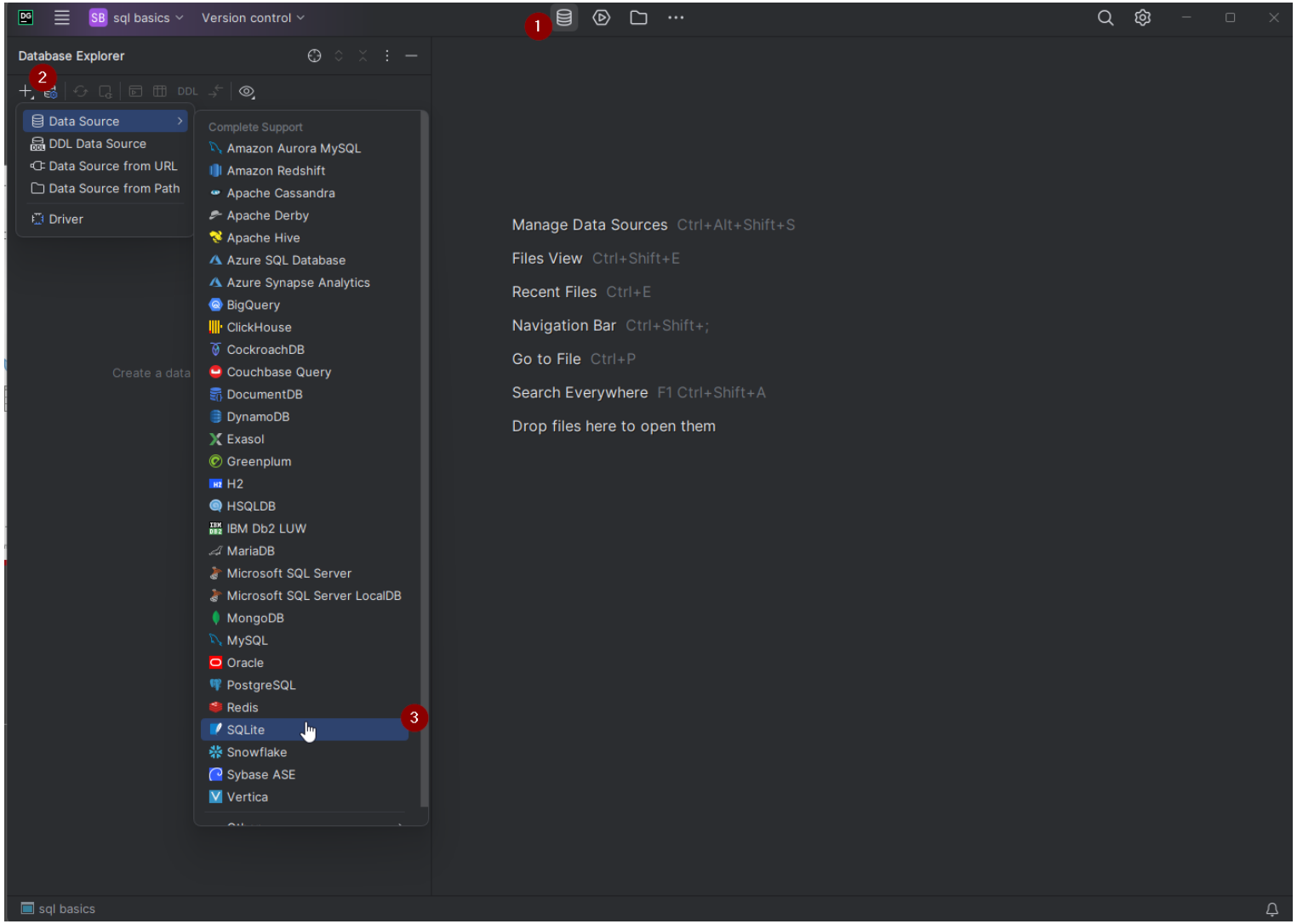
MySQL Types

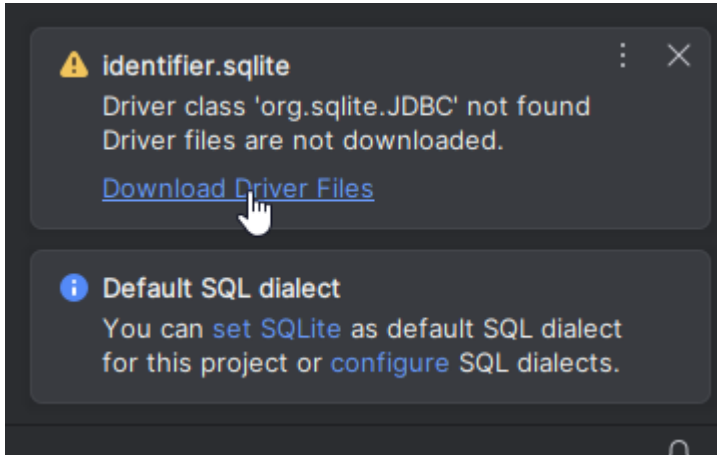
- CHAR(size)
- VARCHAR(size)
- SMALLINT
- INT
- BIGINT
- FLOAT
- DOUBLE
- ...

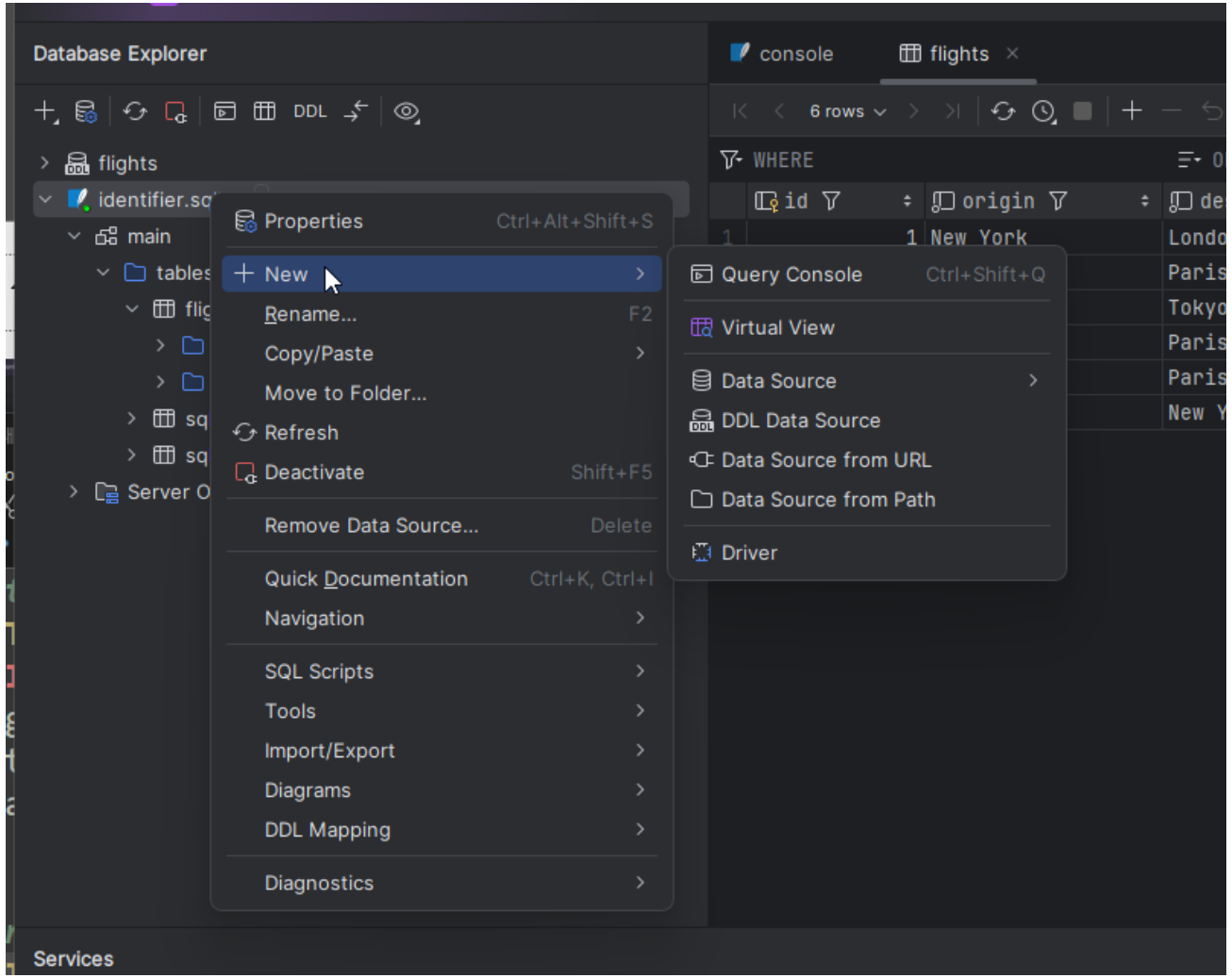
(Optional)



https://www.jetbrains.com/datagrip/download


sql basics
Version control
1
Database Explorer
2
DDL
Data Source
DDL Data Source
Data Source from URL
Data Source from Path
Driver
Complete Support
Amazon Aurora MySQL
Amazon Redshift
Apache Cassandra
Apache Derby
Apache Hive
Azure SQL Database
Azure Synapse Analytics
BigQuery
ClickHouse
CockroachDB
Couchbase Query
DocumentDB
DynamoDB
Exasol
Greenplum
H2
HSQLDB
IBM Db2 LUW
MariaDB
Microsoft SQL Server
Microsoft SQL Server LocalDB
MongoDB
MySQL
Oracle
PostgreSQL
Redis
SQLite
3
Snowflake
Sybase ASE
Vertica
Create a data
Manage Data Sources Ctrl+Alt+Shift+S
Files View Ctrl+Shift+E
Recent Files Ctrl+E
Navigation Bar Ctrl+Shift+;
Go to File Ctrl+P
Search Everywhere F1 Ctrl+Shift+A
Drop files here to open them
sql basics


identifier.sqlite
Driver class 'org.sqlite.JDBC' not found
Driver files are not downloaded.
Download Driver Files
Default SQL dialect
You can set SQLite as default SQL dialect
for this project or configure SQL dialects.


Database Explorer
console
flights
DDL
6 rows
WHERE
flights
identifier.sq
main
tables
Server O
id
origin
1
New York
Properties Ctrl+Alt+Shift+S
New
Rename... F2
Copy/Paste
Move to Folder...
Refresh
Deactivate Shift+F5
Remove Data Source... Delete
Quick Documentation Ctrl+K, Ctrl+I
Navigation
SQL Scripts
Tools
Import/Export
Diagrams
DDL Mapping
Diagnostics
Query Console Ctrl+Shift+Q
Virtual View
Data Source
DDL Data Source
Data Source from URL
Data Source from Path
Driver
Services

CREATE TABLE

```
CREATE TABLE flights (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    origin TEXT NOT NULL,
    destination TEXT NOT NULL,
    duration INTEGER NOT NULL
);
```

```sql
CREATE TABLE flights (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    origin TEXT NOT NULL,
    destination TEXT NOT NULL,
    duration INTEGER NOT NULL
);
```

```sql
CREATE TABLE flights (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    origin TEXT NOT NULL,
    destination TEXT NOT NULL,
    duration INTEGER NOT NULL
);
```

```sql
CREATE TABLE flights (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    origin TEXT NOT NULL,
    destination TEXT NOT NULL,
    duration INTEGER NOT NULL
);
```

```sql
CREATE TABLE flights (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    origin TEXT NOT NULL,
    destination TEXT NOT NULL,
    duration INTEGER NOT NULL
);
```

```sql
CREATE TABLE flights (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    origin TEXT NOT NULL,
    destination TEXT NOT NULL,
    duration INTEGER NOT NULL
);
```

```
CREATE TABLE flights (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    origin TEXT NOT NULL,
    destination TEXT NOT NULL,
    duration INTEGER NOT NULL
);
```

Constraints

- CHECK
- DEFAULT
- NOT NULL
- PRIMARY KEY
- UNIQUE
- ...

INSERT

```
INSERT INTO flights
    (origin, destination, duration)
    VALUES ("New York", "London", 415);
```

```sql
INSERT INTO flights
    (origin, destination, duration)
    VALUES ("New York", "London", 415);
```

```
INSERT INTO flights
    (origin, destination, duration)
    VALUES ("New York", "London", 415);
```

```
INSERT INTO flights
    (origin, destination, duration)
    VALUES ("New York", "London", 415);
```

```
INSERT INTO flights
    (origin, destination, duration)
    VALUES ("New York", "London", 415);
```

```
INSERT INTO flights
    (origin, destination, duration)
    VALUES ("New York", "London", 415);
```

SELECT

```
SELECT * FROM flights;
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT * FROM flights;
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT origin, destination FROM flights;
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT origin, destination FROM flights;
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT * FROM flights WHERE id = 3;
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT * FROM flights WHERE id = 3;
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT * FROM flights WHERE origin = "New York";
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT * FROM flights WHERE origin = "New York";
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT * FROM flights WHERE duration > 500;
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT * FROM flights WHERE duration > 500;
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT * FROM flights WHERE duration > 500
AND destination = "Paris";
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT * FROM flights WHERE duration > 500
AND destination = "Paris";
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT * FROM flights WHERE duration > 500
OR destination = "Paris";
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT * FROM flights WHERE duration > 500
OR destination = "Paris";
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT * FROM flights WHERE
origin IN ("New York", "Lima");
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT * FROM flights WHERE
origin IN ("New York", "Lima");
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT * FROM flights WHERE
origin LIKE "%a%";
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

```
SELECT * FROM flights WHERE
origin LIKE "%a%";
```

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

Functions

- AVERAGE
- COUNT
- MAX
- MIN
- SUM
- ...

UPDATE

```
UPDATE flights
    SET duration = 430
    WHERE origin = "New York"
    AND destination = "London";
```

```
UPDATE flights
    SET duration = 430
    WHERE origin = "New York"
    AND destination = "London";
```

```
UPDATE flights
    SET duration = 430
    WHERE origin = "New York"
    AND destination = "London";
```

```
UPDATE flights
    SET duration = 430
    WHERE origin = "New York"
    AND destination = "London";
```

```
UPDATE flights
    SET duration = 430
    WHERE origin = "New York"
    AND destination = "London";
```

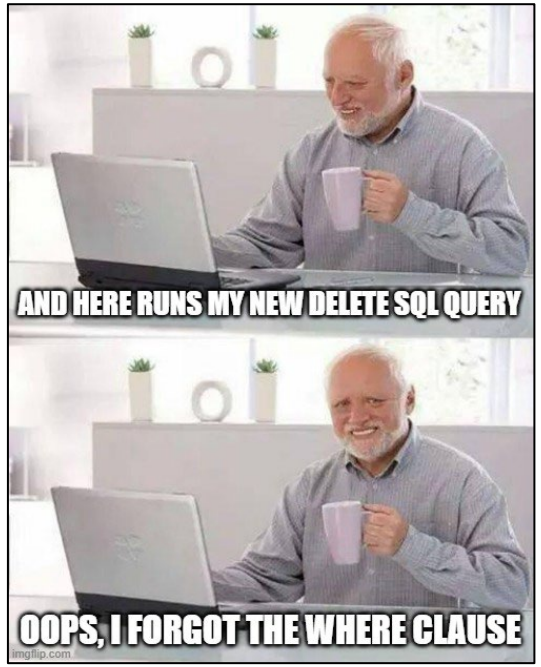
```
UPDATE flights
    SET duration = 430
    WHERE origin = "New York"
    AND destination = "London";
```

DELETE

```
DELETE FROM flights WHERE destination = "Tokyo";
```

```sql
DELETE FROM flights WHERE destination = "Tokyo";
```

```
DELETE FROM flights WHERE destination = "Tokyo";
```




```
DELETE FROM flights WHERE destination = "Tokyo";
```

```
DELETE FROM flights WHERE destination = "Tokyo";
```

Other Clauses

- LIMIT
- ORDER BY
- GROUP BY
- HAVING
- ...

Foreign Keys

flights

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

flights

id	origin	origin_code	destination	destination_code	duration
1	New York	JFK	London	LHR	415
2	Shanghai	PVG	Paris	CDG	760
3	Istanbul	IST	Tokyo	NRT	700
4	New York	JFK	Paris	CDG	435
5	Moscow	SVO	Paris	CDG	245
6	Lima	LIM	New York	JFK	455

airports

id	code	city
1	JFK	New York
2	PVG	Shanghai
3	IST	Istanbul
4	LHR	London
5	SVO	Moscow
6	LIM	Lima
7	CDG	Paris
8	NRT	Tokyo

flights

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

flights

id	origin_id	destination_id	duration
1	1	4	415
2	2	7	760
3	3	8	700
4	1	7	435
5	5	7	245
6	6	1	455

passengers

id	first	last	flight_id
1	Harry	Potter	1
2	Ron	Weasley	1
3	Hermione	Granger	2
4	Draco	Malfoy	4
5	Luna	Lovegood	6
6	Ginny	Weasley	6

people

id	first	last
1	Harry	Potter
2	Ron	Weasley
3	Hermione	Granger
4	Draco	Malfoy
5	Luna	Lovegood
6	Ginny	Weasley

passengers

person_id	flight_id
1	1
2	1
2	4
3	2
4	4
5	6
6	6

JOIN

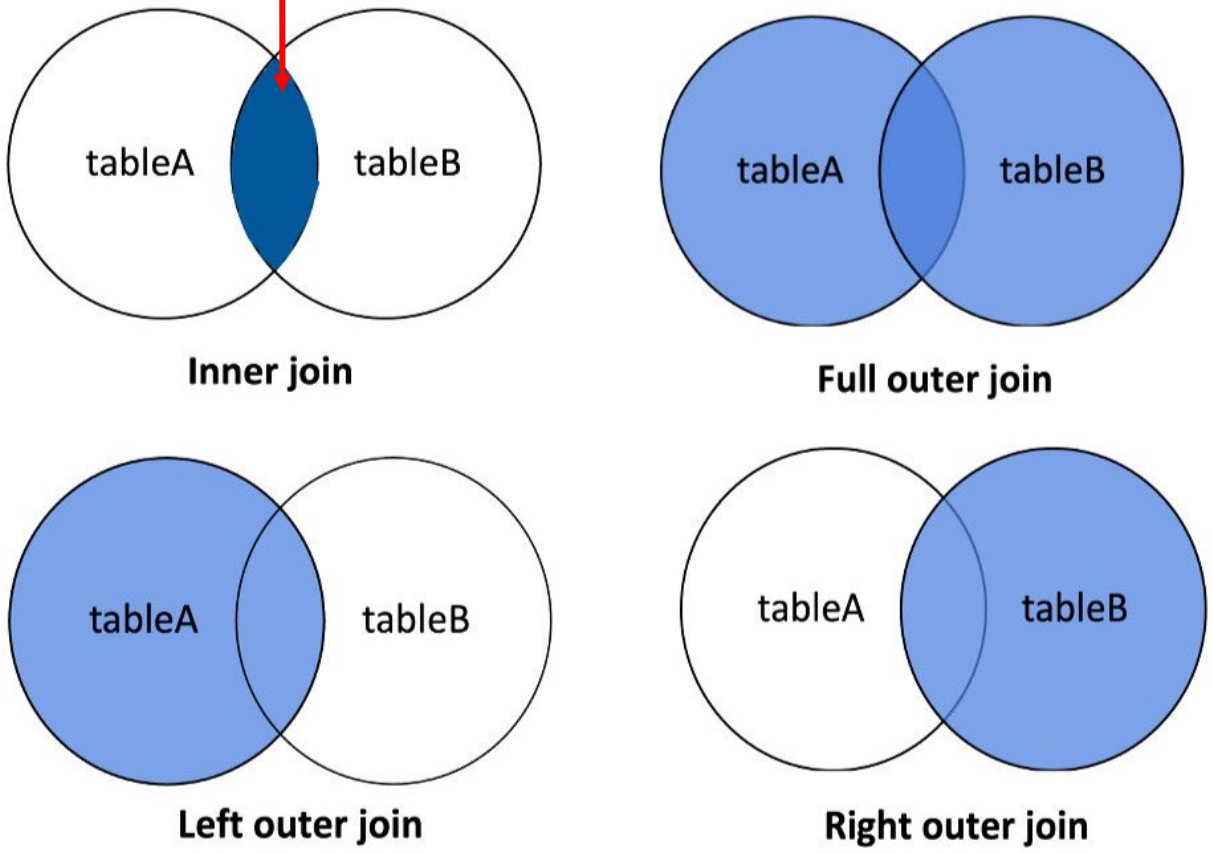
```
SELECT first, origin, destination
FROM flights JOIN passengers
ON passengers.flight_id = flights.id;
```

first	origin	destination
Harry	New York	London
Ron	New York	London
Hermione	Shanghai	Paris
Draco	New York	Paris
Luna	Lima	New York
Ginny	Lima	New York

JOINs

```
ON tableA.id = tableB.id;
```

- JOIN / INNER JOIN
- LEFT OUTER JOIN
- RIGHT OUTER JOIN
- FULL OUTER JOIN



SQL Injection

SEOULTECH
SEOUL NATIONAL UNIVERSITY OF SCIENCE & TECHNOLOGY

Username:

Password:

```
SELECT * FROM users
WHERE username = username AND password = password;
```

Username:

harry

Password:

12345

```
SELECT * FROM users
WHERE username = username AND password = password;
```

```
SELECT * FROM users
WHERE username = "harry" AND password = "12345";
```

Username:

hacker" --

Password:

```
SELECT * FROM users
WHERE username = username AND password = password;
```

```
SELECT * FROM users
WHERE username = "hacker"--" AND password = "";
```

```
SELECT * FROM users
WHERE username = "hacker"--" AND password = "";
```

Race Conditions

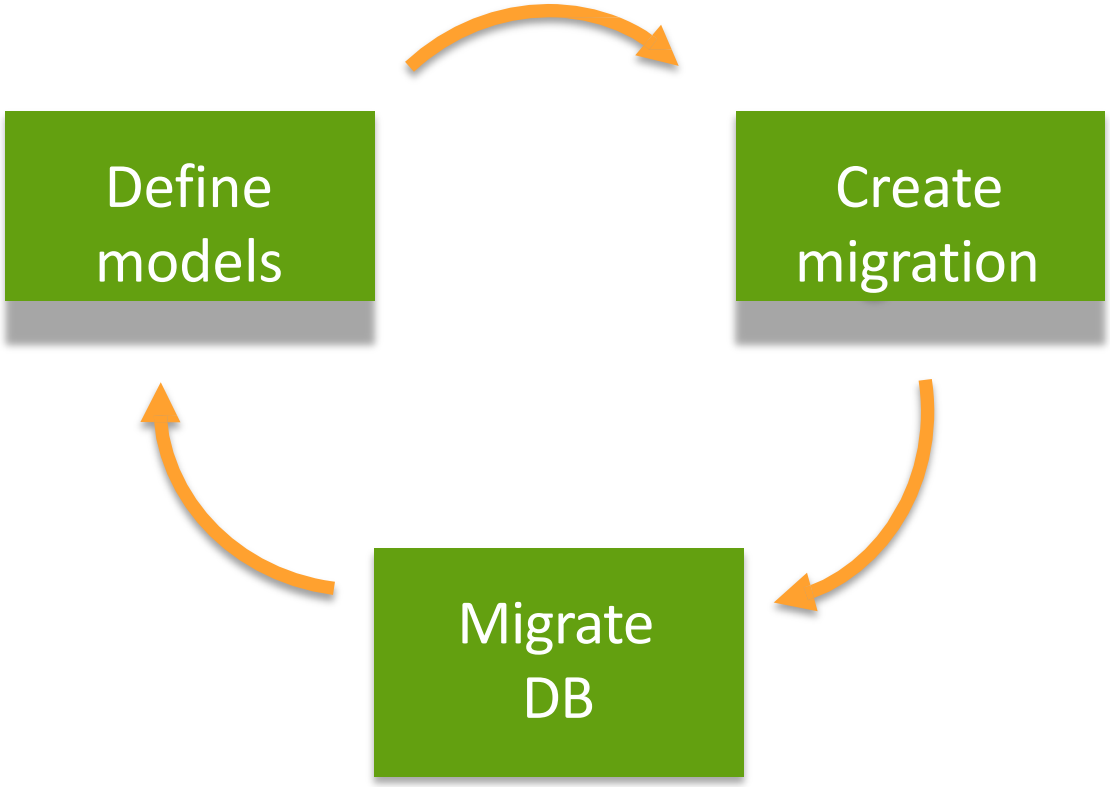
Django ORM

The Django ORM

- **O**bject-**R**elational **M**apping
- Manage your data and database with Python instead of SQL
- Lazy evaluation of queries (database only gets accessed when data is requested, not when forming queries)
- Easy things are easy, medium things are possible, hard things can be done in SQL

Schema Update Workflow

- Define models with fields and relationships
- Create a migration file
- Migrate the database



Update `models.py`

- Create class that subclasses **models.Model**
- Add model fields
- Override __str__ and __repr__

```
from django.db import models

class Flight(models.Model):
    origin = models.CharField(max_length=64)
    destination = models.CharField(max_length=64)
    duration = models.IntegerField()

    def __str__(self):
        return f'{self.id}: {self.origin} to {self.destination}'
```

Model fields

- Lots of Field types
  - BooleanField, CharField, IntegerField, DateTimeField, TextField
  - DecimalField, DurationField, EmailField, UUIDField, + more
  - ImageField, FileField

- Commonly used field arguments
  - null, default, max_length, unique
  - For DateTimeFields
    - auto_now sets to the current datetime when modified
    - auto_now_add sets to the current datetime when created

https://docs.djangoproject.com/en/4.2/ref/models/fields/

Model id's

- When you create a model, an **id** field is automatically created
  - Unless you specify a primary key field
- Default id type is in settings: DEFAULT_AUTO_FIELD
- Access via **object.id** or **object.pk** (pk stands for primary key)
- We often use **pk** (instead of **id**) when retrieving objects, just in case the model uses a custom primary key field
  - e.g. `Flight.objects.get(pk=1)`

Schema Update Workflow

- Define models with fields and relationships
- Create a migration file
  - `$ python manage.py makemigrations`
- Migrate the database
  - `$ python manage.py migrate`

Work with the Django ORM

- Besides in files, you can use the Django ORM in:
  - The terminal via `$ python manage.py shell`

Object creation

- Create a new Flight and save it

```
>>> from flights.models import Flight
>>> f = Flight(origin='New York', destination='London', duration=415)
>>> f.save()
```

Object retrieval

- Get all Flights:

```
Flight.objects.all()
```

- Get a single Visit:

```
f = Flight.objects.first()

f.origin
```

- Filter Airports:

```
>>> airports = Airport.objects.filter(city='Incheon')

>>> airports.count()
```

Relationships

```
class Flight(models.Model):
    origin = models.ForeignKey(Airport,
                               on_delete=models.CASCADE,
                               related_name='departures')
    destination = models.ForeignKey(Airport,
                                    on_delete=models.CASCADE,
                                    related_name='arrivals')
    duration = models.IntegerField()

    def __str__(self):
        return f'{self.id}: {self.origin} to {self.destination}'
```

Relationship field options

- `on_delete` is required
  - CASCADE, SET_NULL, PROTECT are some common values

- `related_name` determines how to get this model from the related one
  - `airport.arrivals.all()`

Defining relationships

- Many-to-one relationship:
  - Use `ForeignKey`
- Many-to-many relationships:
  - Use `ManyToManyField`
- One-to-one relationships:
  - Use `OneToOneField`

Web Programming

The contents of this slide are based or adapted from the content of the video course "Introduction to Django" from Arianne Dee, Pearson 2023.