# Exercise Sheet: Flask-Migrate, Raw SQL, and ORM

This exercise sheet will help you practice database migrations, raw SQL queries, and ORM usage in Flask.

## Part 1: Flask-Migrate

1. Install Flask-Migrate in your virtual environment:

pip install flask-migrate

2. Create a Flask app with the following model:

class Product(db.Model):  
 id = db.Column(db.Integer, primary\_key=True)  
 name = db.Column(db.String(100), nullable=False)  
 price = db.Column(db.Float, nullable=False)

3. Run the migration commands:

- Initialize migrations folder

- Create a migration with message "add product table"

- Upgrade the database

4. Add a new column stock (integer) to the Product model and generate a migration.

## Part 2: Raw SQL Queries

1. Write a raw SQL route to select all products:

SELECT \* FROM product;

2. Write a parameterized raw SQL query to select product by name.

3. Create a raw SQL route to insert a new product with name and price.

4. Write a raw SQL query to update product price by id.

5. Write a raw SQL query to delete a product by id.

## Part 3: ORM Queries

1. Use ORM to fetch all products and return them as JSON.

2. Use ORM to fetch a single product by ID.

3. Use ORM to insert a new product (name, price, stock).

4. Use ORM to update product stock by id.

5. Use ORM to delete a product by id.

## Part 4: Questions

1. What are the main differences between Raw SQL and ORM?

2. When would you prefer Raw SQL over ORM? Give an example.

3. Why is Flask-Migrate important in real-world applications?

4. How can parameterized queries prevent SQL Injection?

5. What are some advantages of using ORM for large projects?