20 Essential SQL Queries — Cheat-Sheet

Big Data & Statistical Learning (FinTech MSc)

Week 1 — Data Sourcing Module

How to read this sheet

Each block shows

- **Pattern**: canonical SQL syntax.

- **Example**: run against the two demo tables you just imported: `spy_daily(date, open, high, low, close, volume)` and `eurusd_daily(date, close)`.

Copy–paste into pgAdmin's *Query Tool*.

---

1. **Select every column**

```
SELECT *
FROM spy_daily;
```

2. **Select specific columns**

```
SELECT date, close
FROM spy_daily;
```

3. **Filter rows with WHERE**

```
SELECT *
FROM spy_daily
WHERE date >= '2025-01-01';
```

4. **Remove duplicates with DISTINCT**

```
SELECT DISTINCT date
FROM spy_daily;
```

5. **Sort results**

```
SELECT date, close
FROM spy_daily
ORDER BY close DESC;
```

6. **Limit number of rows**

```sql
SELECT *
FROM spy_daily
ORDER BY date DESC
LIMIT 10;
```

7. **Aggregate with COUNT()**

```sql
SELECT COUNT(*) AS rows_loaded
FROM spy_daily;
```

8. **GROUP BY summary**

```sql
SELECT DATE_TRUNC('month', date) AS month,
       AVG(close) AS avg_close
FROM spy_daily
GROUP BY 1
ORDER BY 1;
```

9. **Filter aggregates with HAVING**

```sql
SELECT DATE_TRUNC('year', date) AS yr,
       AVG(close) AS avg_close
FROM spy_daily
GROUP BY 1
HAVING AVG(close) > 300;
```

10. **INNER JOIN two tables**

```sql
SELECT s.date,
       s.close AS spy_close,
       e.close AS eurusd_close
FROM spy_daily   s
JOIN eurusd_daily e ON s.date = e.date;
```

11. **LEFT JOIN (keep all left rows)**

```sql
SELECT s.date, s.close, e.close AS fx
FROM spy_daily s
LEFT JOIN eurusd_daily e ON s.date = e.date;
```

12. **RIGHT JOIN**

```sql
SELECT e.date, s.close AS spy, e.close AS fx
FROM spy_daily s
RIGHT JOIN eurusd_daily e ON s.date = e.date;
```

13. **FULL OUTER JOIN**

```
SELECT COALESCE(s.date, e.date) AS date,
       s.close AS spy,
       e.close AS fx
FROM spy_daily s
FULL OUTER JOIN eurusd_daily e ON s.date = e.date;
```

14. **UNION (stack identical columns)**

```
SELECT date, close FROM spy_daily
UNION
SELECT date, close FROM eurusd_daily;
```

15. **INSERT new row**

```
INSERT INTO eurusd_daily(date, close)
VALUES ('2025-07-21', 1.1275);
```

16. **UPDATE existing rows**

```
UPDATE eurusd_daily
SET close = 1.1300
WHERE date = '2025-07-21';
```

17. **DELETE rows**

```
DELETE FROM eurusd_daily
WHERE date < '2000-01-01';
```

18. **CREATE TABLE**

```
CREATE TABLE portfolio (
  id       SERIAL PRIMARY KEY,
  symbol   TEXT,
  quantity NUMERIC,
  cost     NUMERIC
);
```

19. **ALTER TABLE — add column**

```
ALTER TABLE portfolio
ADD COLUMN trade_date DATE;
```

20. **DROP TABLE**

```
DROP TABLE portfolio;
```

Tip: Use EXPLAIN before any query to see how PostgreSQL plans to execute it.