WEEK 1: SQL Data Quality + Python Basics

text

```
DAY 1:
SQL (1 hour):
1. LC 175 - Combine Two Tables (Joins)
2. LC 176 - Second Highest Salary (Ranking)
3. LC 181 - Employees Earning More (Comparisons)

Python (1 hour):
4. LC 1 - Two Sum (Hash maps)
5. LC 26 - Remove Duplicates (Data cleaning)
6. LC 217 - Contains Duplicate (Frequency)

DAY 2:
SQL:
1. LC 182 - Duplicate Emails (Find duplicates)
2. LC 183 - Customers Never Order (Missing data)
3. LC 584 - Find Customer Referee (Validation)

Python:
4. LC 242 - Valid Anagram (String comparison)
5. LC 125 - Valid Palindrome (Text cleaning)
6. LC 344 - Reverse String (Manipulation)

DAY 3:
SQL:
1. LC 196 - Delete Duplicate Emails (Remove duplicates)
2. LC 595 - Big Countries (Filtering)
3. LC 511 - Game Play Analysis I (First events)

Python:
4. LC 20 - Valid Parentheses (Stack validation)
5. LC 121 - Best Time to Buy/Sell (Time series)
6. LC 283 - Move Zeroes (Reorganization)

DAY 4:
SQL:
1. LC 197 - Rising Temperature (Consecutive days)
2. LC 607 - Sales Person (Referential integrity)
```

```
3. LC 586 - Customer Placing Orders (Frequency)

Python:
4. LC 49 - Group Anagrams (Clustering)
5. LC 169 - Majority Element (Statistics)
6. LC 268 - Missing Number (Inference)

DAY 5:
SQL:
1. LC 577 - Employee Bonus (Calculations)
2. LC 610 - Triangle Judgement (Classification)
3. LC 1965 - Employees Missing Info (Completeness)

Python:
4. LC 88 - Merge Sorted Array (Data merging)
5. LC 27 - Remove Element (Filtering)
6. LC 448 - Find Disappeared Numbers (Missing data)

DAY 6 (Review):
SQL: Re-solve 3 hardest from week
Python: Re-solve 3 hardest from week

DAY 7 (Project):
Create: Customer database with cleaning queries

Implement: Data validation scripts in Python
```

WEEK 2: SQL Analytics + Python Statistics

```
text
DAY 8:
SQL:
1. LC 184 - Department Highest Salary (Top performers)
2. LC 185 - Department Top Three (Ranking)
3. LC 178 - Rank Scores (Window functions)

Python:
4. LC 53 - Maximum Subarray (Optimization)
5. LC 152 - Maximum Product Subarray
6. LC 238 - Product Except Self (Feature engineering)
```

DAY 9:
SQL:
1. LC 570 - Managers with 5+ Reports (Hierarchy)
2. LC 574 - Winning Candidate (Voting analytics)
3. LC 1070 - Product Sales III (Time-series)

Python:
4. LC 347 - Top K Frequent (Frequency analysis)
5. LC 215 - Kth Largest Element (Ranking)
6. LC 75 - Sort Colors (Categorization)

DAY 10:
SQL:
1. LC 1084 - Sales Analysis III (Product performance)
2. LC 550 - Game Play Analysis IV (Retention)
3. LC 262 - Trips and Users (Rate calculation)

Python:
4. LC 56 - Merge Intervals (Range consolidation)
5. LC 435 - Non-overlapping Intervals
6. LC 406 - Queue Reconstruction

DAY 11:
SQL:
1. LC 569 - Median Employee Salary (Statistics)
2. LC 571 - Weighted Median
3. LC 579 - Cumulative Salary (Running totals)

Python:
4. LC 229 - Majority Element II (Thresholds)
5. LC 384 - Shuffle Array (Random sampling)
6. LC 398 - Random Pick Index

DAY 12:
SQL:
1. LC 615 - Avg Salary: Dept vs Company (Comparative stats)
2. LC 618 - Students Report By Geography (Pivot)
3. LC 1225 - Report Contiguous Dates (Continuity)

Python:
4. LC 528 - Random Pick with Weight (Probability)
5. LC 973 - K Closest Points (Distance metrics)

```
6. LC 349 - Intersection of Arrays

DAY 13 (Review):
SQL: Re-solve window function problems
Python: Re-solve statistical problems

DAY 14 (Project):
Analyze: Sales data with SQL window functions
Calculate: Statistical metrics in Python from scratch
```

WEEK 3: SQL Time Series + Python Time Analysis

text

```
DAY 15:
SQL:
1. LC 1097 - Game Play Analysis V (Install date)
2. LC 1126 - Active Businesses (Thresholds)
3. LC 1158 - Market Analysis I (First purchase)

Python:
4. LC 122 - Best Time to Buy/Sell II
5. LC 309 - With Cooldown
6. LC 714 - With Transaction Fee

DAY 16:
SQL:
1. LC 1321 - Restaurant Growth (Moving averages)
2. LC 1532 - Most Recent Three Orders (Recency)
3. LC 1596 - Most Frequent Products

Python:
4. LC 300 - Longest Increasing Subsequence
5. LC 198 - House Robber (Optimization)
6. LC 213 - House Robber II

DAY 17:
SQL:
1. LC 1709 - Biggest Window Between Visits (Gap analysis)
2. LC 1892 - Page Recommendations II (Behavioral)
3. LC 1045 - Customers Bought All (Segmentation)
```

```
Python:
4. LC 188 - Best Time to Buy/Sell IV
5. LC 322 - Coin Change
6. LC 416 - Partition Equal Subset

DAY 18:
SQL:
1. LC 1107 - New Users Daily Count (Acquisition)
2. LC 1112 - Highest Grade Each Student (Performance)
3. LC 1132 - Reported Posts II (Engagement)

Python:
4. LC 11 - Container With Most Water
5. LC 15 - 3Sum
6. LC 16 - 3Sum Closest

DAY 19:
SQL:
1. LC 1149 - Article Views II (Content analytics)
2. LC 1159 - Market Analysis II (Preferences)
3. LC 1082 - Sales Analysis I (Best sellers)

Python:
4. LC 42 - Trapping Rain Water
5. LC 239 - Sliding Window Maximum
6. LC 295 - Find Median from Stream

DAY 20 (Review):
SQL: Time series patterns review
Python: Dynamic programming review

DAY 21 (Project):
Build: Time series forecasting with SQL features
Implement: Stock analysis in Python
```

WEEK 4: SQL Complex + Python Matrix/Text

text

```
DAY 22:
```

SQL:
1. LC 601 - Human Traffic of Stadium (Complex windows)
2. LC 626 - Exchange Seats
3. LC 1635 - Hopper Queries I (Advanced analytics)

Python:
4. LC 48 - Rotate Image (Matrix)
5. LC 54 - Spiral Matrix
6. LC 73 - Set Matrix Zeroes

DAY 23:
SQL:
1. LC 1645 - Hopper Queries II
2. LC 1651 - Hopper Queries III
3. LC 1393 - Capital Gain/Loss (Performance)

Python:
4. LC 74 - Search 2D Matrix
5. LC 240 - Search 2D Matrix II
6. LC 378 - Kth Smallest in Matrix

DAY 24:
SQL:
1. LC 1501 - Safe Investment Countries (Success metrics)
2. LC 1747 - Leetflex Banned (Anomaly detection)
3. LC 1809 - Ad-Free Sessions (A/B testing)

Python:
4. LC 3 - Longest Substring Without Repeating
5. LC 5 - Longest Palindromic Substring
6. LC 76 - Minimum Window Substring

DAY 25:
SQL:
1. LC 1831 - Max Transaction Daily (Peak detection)
2. LC 2084 - Drop Type 1 Orders (Business rules)
3. LC 1179 - Reformat Department Table (Pivot)

Python:
4. LC 438 - Find All Anagrams
5. LC 647 - Palindromic Substrings
6. LC 680 - Valid Palindrome II

```
DAY 26:
SQL:
1. LC 1270 - All Report to Manager (Hierarchy)
2. LC 1767 - Subtasks Not Executed
3. LC 1412 - Quiet Students (Complex filtering)

Python:
4. LC 151 - Reverse Words
5. LC 387 - First Unique Character
6. LC 383 - Ransom Note

DAY 27 (Review):
SQL: Complex query patterns
Python: Text processing techniques

DAY 28 (Project):
Create: Business dashboard with SQL aggregates
Build: Text analysis pipeline in Python
```

WEEK 5: MONTH 1 REVIEW & INTEGRATION

text

```
DAY 29:
SQL Speed Run (10 problems in 90 mins):
1. LC 182, 184, 185, 197, 262
2. LC 569, 570, 601, 1070, 1321

Python Speed Run (10 problems in 90 mins):
1. LC 1, 3, 26, 49, 121
2. LC 238, 283, 347, 438, 560

DAY 30:
Weak Areas Focus:
- Identify 5 weakest SQL topics
- Identify 5 weakest Python topics
- Master each with 3 similar problems
- Create cheat sheets
```

🚀 MONTH 2: ADVANCED APPLICATIONS (30 Days)

WEEK 6: SQL Feature Engineering + Python Graphs

text

```
DAY 31:
SQL - Temporal Features:
1. Create: Date difference features
2. Create: Day of week/month features
3. Create: Holiday flag features
4. Create: Seasonality features
5. Create: Time bucket features

Python:
6. LC 200 - Number of Islands (Graph)
7. LC 207 - Course Schedule (Dependencies)
8. LC 210 - Course Schedule II

DAY 32:
SQL - Aggregated Features:
1. Create: Rolling averages (7, 30, 90 day)
2. Create: Cumulative sums
3. Create: Percent changes
4. Create: Z-scores
5. Create: Rank features

Python:
6. LC 133 - Clone Graph
7. LC 261 - Graph Valid Tree
8. LC 323 - Connected Components

DAY 33:
SQL - Window Feature Patterns:
1. LC 197 - Rising Temperature (pattern expansion)
2. LC 550 - Retention features expansion
3. LC 1321 - Moving features variations
4. Create: Lead/lag difference features
```

5. Create: First/last occurrence features

Python:
6. LC 547 - Number of Provinces
7. LC 695 - Max Area of Island
8. LC 994 - Rotting Oranges

DAY 34:
SQL - Cohort Analysis Implementation:
1. Create: User acquisition cohorts
2. Create: Retention rate by cohort
3. Create: LTV calculation by cohort
4. Create: Cohort behavior trends
5. Create: Cross-cohort comparisons

Python:
6. LC 417 - Pacific Atlantic Water Flow
7. LC 529 - Minesweeper
8. LC 684 - Redundant Connection

DAY 35:
SQL - Business Metric Features:
1. Create: RFM scores (Recency, Frequency, Monetary)
2. Create: Churn risk scores
3. Create: Customer segmentation features
4. Create: Product affinity scores
5. Create: Price elasticity features

Python:
6. LC 785 - Is Graph Bipartite
7. LC 797 - All Paths From Source to Target
8. LC 841 - Keys and Rooms

DAY 36 (Review):
SQL: Feature engineering patterns
Python: Graph traversal methods

DAY 37 (Project):
Build: Complete feature engineering pipeline

Implement: Network analysis on user data

WEEK 7: SQL Performance + Python Optimization

```text
DAY 38:
SQL Optimization I:
1. Analyze: EXPLAIN plans for 5 complex queries
2. Create: Strategic indexes
3. Rewrite: Subqueries as joins
4. Optimize: Window function queries
5. Test: Large dataset performance

Python:
6. LC 146 - LRU Cache
7. LC 460 - LFU Cache
8. LC 588 - In-Memory File System

DAY 39:
SQL Optimization II:
1. Partition: Large tables strategy
2. Materialize: Common results
3. Optimize: String operations
4. Improve: Date calculations
5. Benchmark: Before/after improvements

Python:
9. LC 295 - Median from Data Stream (optimized)
10. LC 480 - Sliding Window Median
11. LC 703 - Kth Largest in Stream

DAY 40:
SQL Query Patterns for Big Data:
1. Create: Chunked processing queries
2. Implement: Sampling strategies
3. Design: Incremental loading
4. Handle: Memory constraints
5. Optimize: Join order for large tables

Python:
6. LC 23 - Merge K Sorted Lists
7. LC 76 - Minimum Window Substring (optimized)
8. LC 239 - Sliding Window Max (optimized)
```

```
DAY 41:
SQL Security & Best Practices:
1. Implement: SQL injection prevention
2. Create: Role-based access queries
3. Design: Audit logging
4. Handle: Sensitive data masking
5. Implement: Data validation at DB level

Python:
6. LC 381 - Insert Delete GetRandom O(1) Duplicates
7. LC 432 - All O(1) Data Structure
8. LC 460 - LFU Cache (re-implement)

DAY 42:
SQL Advanced CTEs:
1. LC 1384 - Total Sales by Year
2. LC 1613 - Find Missing IDs
3. Create: Recursive organization hierarchy
4. Implement: Graph traversal in SQL
5. Solve: Complex business logic with CTEs

Python:
6. LC 84 - Largest Rectangle in Histogram
7. LC 85 - Maximal Rectangle
8. LC 316 - Remove Duplicate Letters

DAY 43 (Review):
SQL: Performance optimization techniques
Python: System design patterns

DAY 44 (Project):
Optimize: Slow database queries
```

```
Implement: Caching system in Python
```

WEEK 8: Real-world Scenarios

```
text
```

```
DAY 45:
UK NHS Analysis:
```

SQL:
1. Patient wait time trends
2. Treatment outcome analysis
3. Resource utilization
4. Readmission prediction features

Python:
5. Medical statistics calculations
6. Patient clustering
7. Risk score implementation

DAY 46:
UK Retail Analytics:
SQL:
1. Customer segmentation (RFM)
2. Sales forecasting features
3. Inventory optimization
4. Promotion effectiveness

Python:
5. Demand forecasting model
6. Price optimization algorithm
7. Recommendation system basics

DAY 47:
UK Finance Analytics:
SQL:
1. Fraud detection patterns
2. Credit risk scoring
3. Portfolio optimization
4. Regulatory reporting

Python:
5. Financial time series analysis
6. Risk metric calculations
7. Monte Carlo simulations

DAY 48:
E-commerce Analytics:
SQL:
1. User journey analysis
2. Conversion funnel metrics

```
3. A/B test results analysis
4. Customer lifetime value

Python:
5. Cohort retention analysis
6. Churn prediction features
7. Personalization algorithms

DAY 49:
Social Media Analytics:
SQL:
1. Engagement metrics
2. Network analysis queries
3. Content performance
4. User growth analysis

Python:
5. Graph analysis algorithms
6. Sentiment analysis basics
7. Viral content prediction

DAY 50 (Review):
Industry-specific patterns
Cross-domain applications

DAY 51 (Project):
Choose: One industry domain
Build: Complete analytics solution
Present: Business insights
```

WEEK 9: Mock Interviews & Integration

text

```
DAY 52:
Mock Interview 1 (SQL):
1. LC 185 - Department Top Three
2. LC 569 - Median Employee Salary
3. Business case: Sales analytics

Mock Interview 1 (Python):
```

4. LC 239 - Sliding Window Maximum
5. LC 295 - Median from Data Stream
6. Data processing task

DAY 53:
Mock Interview 2 (SQL):
1. LC 601 - Human Traffic
2. LC 1321 - Restaurant Growth
3. Business case: User retention

Mock Interview 2 (Python):
4. LC 56 - Merge Intervals
5. LC 238 - Product Except Self
6. Statistical analysis task

DAY 54:
Mock Interview 3 (SQL):
1. LC 550 - Game Play Analysis IV
2. LC 1097 - Game Play Analysis V
3. Business case: Time series analysis

Mock Interview 3 (Python):
4. LC 3 - Longest Substring
5. LC 49 - Group Anagrams
6. Text processing task

DAY 55:
Mock Interview 4 (SQL):
1. LC 262 - Trips and Users
2. LC 1709 - Biggest Window
3. Business case: Rate calculations

Mock Interview 4 (Python):
4. LC 42 - Trapping Rain Water
5. LC 11 - Container With Most Water
6. Optimization task

DAY 56:
Mock Interview 5 (SQL):
1. LC 1635 - Hopper Queries
2. LC 1767 - Subtasks Not Executed
3. Business case: Complex analytics

```
Mock Interview 5 (Python):
4. LC 84 - Largest Rectangle
5. LC 85 - Maximal Rectangle
6. Advanced algorithm task

DAY 57 (Review):
Interview performance analysis
Common mistakes identification
Improvement strategies

DAY 58 (Project):
Take-home assignment simulation
Complete end-to-end analysis
```

```
Professional documentation
```

WEEK 10: MONTH 2 REVIEW & POLISH

```
text
```

```
DAY 59:
SQL Mastery Test:
- Solve 15 problems in 3 hours
- Mix of all difficulty levels
- Production-quality code
- Optimization considerations

DAY 60:
Python Mastery Test:
- Solve 15 problems in 3 hours
- Mix of all difficulty levels
- Clean, efficient code
- Comprehensive testing

DAY 61:
Weak Areas Deep Dive:
- Identify remaining gaps
- Master with focused practice
- Create detailed notes
- Build mental models
```

```
DAY 62:
Speed Training:
- Easy: 5 mins/problem
- Medium: 15 mins/problem
- Hard: 25 mins/problem
- Track success rates

DAY 63:
Pattern Recognition:
- Group problems by pattern
- Create solution templates
- Develop recognition speed
- Build intuition

DAY 64:
Final Mock Interview:
- 3-hour comprehensive test
- SQL + Python + Business case
- Record and analyze
- Final adjustments

DAY 65:
Rest & Mental Preparation:
- Light review only
- Success visualization
- Interview mindset
- Confidence building
```

🎯 MONTH 3: SPECIALIZATION & INTERVIEW READY (30 Days)

WEEK 11: Finance Track (Choose Your Specialization)

```text
DAY 66:
SQL Finance I:
```

1. Portfolio performance attribution
2. Risk exposure calculations
3. VaR (Value at Risk) simulations
4. Sharpe ratio calculations
5. Correlation analysis

Python Finance I:

6. Stock price analysis
7. Return volatility calculations
8. Moving average strategies
9. Technical indicators
10. Backtesting framework

DAY 67:

SQL Finance II:

1. Customer lifetime value
2. Credit scoring models
3. Fraud detection patterns
4. Regulatory capital requirements
5. Stress testing scenarios

Python Finance II:

6. Monte Carlo simulations
7. Option pricing models
8. Risk metric calculations
9. Portfolio optimization
10. Time series forecasting

DAY 68:

SQL Finance III:

1. Trading pattern analysis
2. Market impact measurements
3. Liquidity analysis
4. Transaction cost analysis
5. Compliance reporting

Python Finance III:

6. Algorithmic trading basics
7. Order book analysis
8. Market microstructure
9. High-frequency data
10. Real-time analytics

```
DAY 69-72: Finance Project

Build complete financial analytics platform
```

WEEK 12: Tech/E-commerce Track

```text
DAY 73:
SQL Tech I:
1. User behavior analytics
2. A/B test analysis
3. Conversion funnel metrics
4. Engagement measurements
5. Growth metrics

Python Tech I:
6. Recommendation algorithms
7. Collaborative filtering
8. Content-based filtering
9. Ranking algorithms
10. Personalization systems

DAY 74:
SQL Tech II:
1. System performance metrics
2. Error rate analysis
3. Latency measurements
4. Capacity planning
5. Resource utilization

Python Tech II:
6. API rate limiting
7. Caching strategies
8. Load balancing simulation
9. Distributed systems basics
10. Scalability patterns

DAY 75:
SQL Tech III:
1. Product analytics
```

```
2. Feature usage tracking
3. Customer feedback analysis
4. Competitor analysis
5. Market positioning

Python Tech III:
6. Natural language processing basics
7. Sentiment analysis
8. Topic modeling
9. Search algorithms
10. Information retrieval

DAY 76-79: Tech Project
```

```
Build complete tech analytics solution
```

WEEK 13: Healthcare/Public Sector Track

text

```
DAY 80:
SQL Healthcare I:
1. Patient outcome analysis
2. Treatment effectiveness
3. Resource allocation
4. Wait time optimization
5. Readmission rates

Python Healthcare I:
6. Medical statistics
7. Survival analysis basics
8. Risk stratification
9. Clinical decision support
10. Epidemiology models

DAY 81:
SQL Healthcare II:
1. Population health metrics
2. Disease prevalence
3. Healthcare access analysis
4. Cost effectiveness
5. Quality measures
```

```
Python Healthcare II:
6. Medical image analysis basics
7. Signal processing
8. Time series medical data
9. Predictive modeling
10. Anomaly detection

DAY 82:
SQL Public Sector:
1. Crime pattern analysis
2. Education outcomes
3. Transportation analytics
4. Environmental data
5. Economic indicators

Python Public Sector:
6. Geospatial analysis
7. Demographic analysis
8. Policy impact assessment
9. Resource optimization
10. Public service analytics

DAY 83-86: Healthcare Project
```

```
Build healthcare analytics system
```

WEEK 14: Final Preparation

```
text
```

```
DAY 87:
Company-Specific Preparation:
- Research target companies
- Study their interview patterns
- Practice their specific problems
- Understand their tech stack
- Learn their business domain

DAY 88:
Behavioral & Communication:
- Practice explaining solutions
- Develop storytelling skills
- Prepare for case studies
- Practice whiteboarding
- Improve presentation skills

DAY 89:
Portfolio & Resume:
- Update GitHub with all projects
- Polish portfolio website
- Tailor resume for each application
- Prepare cover letter templates
- Gather references

DAY 90:
Final Review & Strategy:
- Top 50 problems quick review
- Common pitfalls reminder
- Time management strategies
- Stress management techniques
- Success visualization
```

📊 DAILY EXECUTION TEMPLATE

2-Hour Daily Structure:

text

```
18:00-18:10: Warm-up (review yesterday)
18:10-18:40: SQL problems (30 mins)
18:40-19:10: Python problems (30 mins)
19:10-19:40: Review solutions (30 mins)
19:40-20:00: Planning & notes (20 mins)
```

Weekend Boost (Saturday):

```
text
10:00-12:00: Project work
14:00-16:00: Mock interview
16:00-17:00: Review & planning
```

Weekly Review (Sunday):

```
text
10:00-11:00: Progress assessment
11:00-12:00: Plan next week
14:00-15:00: Network/learn
```

🎯 SUCCESS METRICS

Daily Targets:

- ✅ 3 SQL problems solved
- ✅ 3 Python problems solved
- ✅ Solutions optimized
- ✅ Notes taken
- ✅ GitHub updated

Weekly Targets:

- ✅ 21 problems each (SQL + Python)
- ✅ 1 project progressed
- ✅ 1 mock interview completed
- ✅ Skills improved in weak areas
- ✅ Network expanded

Monthly Targets:

- ✅ 90 problems each (SQL + Python)
- ✅ 4 projects completed
- ✅ 4 mock interviews mastered
- ✅ Portfolio enhanced
- ✅ Interview ready

🚀 START TOMORROW

Day 1 (Tomorrow):

```
text

SQL (60 mins):
1. LC 175 - Combine Two Tables (15 mins)
2. LC 176 - Second Highest Salary (20 mins)
3. LC 181 - Employees Earning More (25 mins)

Python (60 mins):
4. LC 1 - Two Sum (20 mins)
5. LC 26 - Remove Duplicates (20 mins)

6. LC 217 - Contains Duplicate (20 mins)
```

Tools Ready Tonight:

1. LeetCode account (free tier works)
2. PostgreSQL installed locally
3. Python 3.8+ with VS Code
4. GitHub repository created
5. Progress tracker set up

💡 ADJUSTMENT OPTIONS

If Time Limited:

```
text
Option A: Reduce to 3 problems/day (1.5 SQL + 1.5 Python)
Option B: Focus on priority problems only
Option C: Extend timeline to 4 months
```

If Advanced:

```
text
Option A: Increase to 4 problems each/day
Option B: Add system design problems
Option C: Start mock interviews earlier
```

If Beginner:

```
text
Option A: Start with easier problems
Option B: Spend more time on fundamentals
Option C: Reduce daily target to 2+2 problems
```

Total Commitment: 2 hours/day × 90 days = 180 hours

Total Problems: 270 SQL + 270 Python = 540 problems

Projects: 12+ complete analytics projects

Outcome: UK Data Science interview ready