Crypto Data Database Tasks

Setup Tasks

Database Creation

- [ ] Create crypto_data database
- [ ] Set up database user permissions
- [ ] Configure database connection parameters
- [ ] Set up database backup procedures
- [ ] Test database connectivity

Table Creation

- [ ] Create price_data table
- [ ] Create market_data table
- [ ] Create defi_protocols table
- [ ] Create exchange_data table
- [ ] Create blockchain_metrics table
- [ ] Create whale_transactions table

Index Optimization

- [ ] Create symbol-timestamp indexes
- [ ] Create performance indexes
- [ ] Create unique constraint indexes
- [ ] Optimize query performance indexes
- [ ] Validate index effectiveness

View Creation

- [ ] Create latest_prices view
- [ ] Create top_cryptos view
- [ ] Create top_defi_protocols view
- [ ] Create recent_whale_activity view
- [ ] Create market_overview view

Testing Tasks

Connection Tests

- [ ] Test database connectivity
- [ ] Validate user permissions
- [ ] Test connection pooling
- [ ] Verify SSL connections
- [ ] Test failover mechanisms

Data Integrity Tests

- [ ] Test primary key constraints

- [ ] Validate foreign key relationships
- [ ] Test unique constraints
- [ ] Verify data type constraints
- [ ] Test null value handling

Performance Tests

- [ ] Measure query response times
- [ ] Test concurrent connections
- [ ] Validate index performance
- [ ] Test large dataset queries
- [ ] Monitor memory usage

Data Validation Tests

- [ ] Test data insertion accuracy
- [ ] Validate data update operations
- [ ] Test data deletion procedures
- [ ] Verify data consistency
- [ ] Test transaction rollbacks

Configuration Tasks

Database Optimization

- [ ] Configure memory settings
- [ ] Optimize query cache
- [ ] Set up connection pooling
- [ ] Configure backup procedures
- [ ] Set up monitoring

Security Configuration

- [ ] Set up user access controls
- [ ] Configure SSL encryption
- [ ] Set up audit logging
- [ ] Implement data encryption
- [ ] Configure firewall rules

Data Management Tasks

Data Loading

- [ ] Set up data ingestion pipelines
- [ ] Configure real-time data feeds
- [ ] Set up batch data processing
- [ ] Implement data validation rules
- [ ] Set up error handling

Data Maintenance

- [ ] Set up data archiving

- [ ] Configure data retention policies
- [ ] Set up data cleanup procedures
- [ ] Implement data quality checks
- [ ] Set up data monitoring

Optimization Tasks

Performance Tuning

- [ ] Optimize slow queries
- [ ] Implement query caching
- [ ] Configure database partitioning
- [ ] Set up read replicas
- [ ] Monitor resource usage

Monitoring Setup

- [ ] Set up database health checks
- [ ] Configure performance metrics
- [ ] Set up error tracking
- [ ] Create usage dashboards
- [ ] Set up alerting

Documentation Tasks

Technical Documentation

- [ ] Document database schema
- [ ] Create query examples
- [ ] Document stored procedures
- [ ] Create troubleshooting guide
- [ ] Document best practices

User Documentation

- [ ] Create setup guide
- [ ] Document common queries
- [ ] Create FAQ section
- [ ] Document limitations
- [ ] Create migration guide

Completion Criteria

Functional Requirements

- [ ] Database connects successfully
- [ ] All tables created correctly
- [ ] Views return accurate data
- [ ] Indexes improve performance
- [ ] Data integrity maintained

Non-Functional Requirements

- [ ] Query response time < 5 seconds
- [ ] 99.9% uptime achieved
- [ ] Data consistency maintained
- [ ] Security requirements met
- [ ] Backup procedures tested

Validation Checklist

Pre-Production

- [ ] All tests passing
- [ ] Performance benchmarks met
- [ ] Security scan completed
- [ ] Documentation reviewed
- [ ] Monitoring configured

Production Ready

- [ ] Load testing completed
- [ ] Disaster recovery tested
- [ ] Monitoring alerts configured
- [ ] Support procedures documented
- [ ] Rollback procedures tested

Success Metrics

- **Uptime**: > 99.9%
- **Query Performance**: < 5 seconds average
- **Data Accuracy**: > 99.9%
- **Storage Efficiency**: > 80%
- **Backup Success**: 100%

Known Issues & Limitations

- Large datasets may require partitioning
- Real-time updates need careful indexing
- Historical data storage grows rapidly
- Complex queries may need optimization
- Concurrent access requires proper locking

Support & Troubleshooting

Common Issues

1. **Slow Queries**: Check indexes and query optimization
2. **Connection Issues**: Verify network and credentials
3. **Data Inconsistency**: Check transaction isolation
4. **Storage Issues**: Monitor disk space and archiving

5. **Performance Degradation**: Analyze query patterns

Escalation Path

1. Check database logs and metrics
2. Consult troubleshooting guide
3. Contact database administrator
4. Escalate to infrastructure team
5. Create incident ticket if needed