MCP Test Commands Reference

This document contains execute commands and natural language prompts for all MCP methods.

Contract IDs Quick Reference

MCP	Contract ID
Anomaly Detection	`aaaaaay3luxuojcss2b7lptxckmhydqkatcinoiogzuep66itps6ntv7pe`
Entity Relationship	`aaaaaa5nb5t5cgg7nrintzf56dpmb3otezf3hycr47mbwocei5zwm5h4gm`
Trade Data	`aaaaaa7bmmtxibdkvxut2rpyuplk57gadst7qlz5hkl7ugalw6co4lsryy`
UPSI Database	`aaaaaaytejhz4sy6rhqobwg2sqvloy5276ouvo6ejod2fzt4rtobggxtf4`
Jira MCP	`aaaaaa22qx66rbpoh6ak2mmjfb5kc2ryjnoqzp33ujzi73u3aze5wkij7e`

1. Anomaly Detection MCP

Contract ID: `aaaaaay3luxuojcss2b7lptxckmhydqkatcinoiogzuep66itps6ntv7pe`

get_context

Natural Language: "Show me what entities and symbols were recently queried"

```
execute -n aaaaaay3luxuojcss2b7lptxckmhydqkatcinoiogzuep66itps6ntv7pe -m get_context
```

detect_spoofing

Natural Language: "Check if order ORD123 from trader ENT-REL-001 on RELIANCE is spoofing"

```
execute -n aaaaaay3luxuojcss2b7lptxckmhydqkatcinoiogzuep66itps6ntv7pe -m detect_spoofing -i '{"order_id":"ORD123","entity_id":"ENT-
```

detect_wash_trading

Natural Language: "Check if ENT-REL-001 and ACC017 are doing wash trades on IBM"

```
execute -n aaaaaay3luxuojcss2b7lptxckmhydqkatcinoiogzuep66itps6ntv7pe -m detect_wash_trading -i '{"entity_id":"ENT-REL-001","counter
```

detect_pump_dump

Natural Language: "Is there a pump and dump happening on AAPL in the last 60 minutes?"

```
execute -n aaaaaay3luxuojcss2b7lptxckmhydqkatcinoiogzuep66itps6ntv7pe -m detect_pump_dump -i '{"symbol":"AAPL","time_window_minutes"
```

detect_front_running

Natural Language: "Check if broker BRK-001 front-ran a client trade on MSFT"

```
execute -n aaaaaay3luxuojcss2b7lptxckmhydqkatcinoiogzuep66itps6ntv7pe -m detect_front_running -i '{"entity_id":"BRK-001","symbol":"
```

analyze_volume_anomaly

Natural Language: "Analyze volume anomalies for IBM on 15min intervals"

```
execute -n aaaaaay3luxuojcss2b7lptxckmhydqkatcinoiogzuep66itps6ntv7pe -m analyze_volume_anomaly -i '{"symbol":"IBM","interval":"15mi
```

check_rsi_levels

Natural Language: "What are the RSI levels for GOOGL?"

```
execute -n aaaaaay3luxuojcss2b7lptxckmhydqkatcinoiogzuep66itps6ntv7pe -m check_rsi_levels -i '{"symbol":"GOOGL"}'
```

scan_entity_anomalies

Natural Language: "Run a full anomaly scan on suspect SUS-001"

```
execute -n aaaaaay3luxuojcss2b7lptxckmhydqkatcinoiogzuep66itps6ntv7pe -m scan_entity_anomalies -i '{"entity_id":"SUS-001"}'
```

tools

Natural Language: "What methods are available in anomaly detection?"

```
execute -n aaaaaay3luxuojcss2b7lptxckmhydqkatcinoiogzuep66itps6ntv7pe -m tools
```

2. Entity Relationship MCP

Contract ID: `aaaaaa5nb5t5cgg7nrintzf56dpmb3otezf3hycr47mbwocei5zwm5h4gm`

get_context

Natural Language: "Show me the recently queried entities and companies"

```
execute -n aaaaaa5nb5t5cgg7nrintzf56dpmb3otezf3hycr47mbwocei5zwm5h4gm -m get_context
```

get_entity

Natural Language: "Get details for Mukesh Ambani (ENT-REL-001)"

```
execute -n aaaaaa5nb5t5cgg7nrintzf56dpmb3otezf3hycr47mbwocei5zwm5h4gm -m get_entity -i '{"entity_id":"ENT-REL-001"}'
```

Natural Language (Fuzzy): "Get details for entity REL-001"

```
execute -n aaaaaa5nb5t5cgg7nrintzf56dpmb3otezf3hycr47mbwocei5zwm5h4gm -m get_entity -i '{"entity_id":"REL-001"}'
```

search_entities

Natural Language: "Search for entities with name containing Ambani"

```
execute -n aaaaaa5nb5t5cgg7nrintzf56dpmb3otezf3hycr47mbwocei5zwm5h4gm -m search_entities -i '{"search_query":"Ambani","limit":5}'
```

Natural Language: "Find entities by PAN AABPA1234A"

```
execute -n aaaaaa5nb5t5cgg7nrintzf56dpmb3otezf3hycr47mbwocei5zwm5h4gm -m search_entities -i '{"search_query":"AABPA1234A","limit":5
```

get_relationships

Natural Language: "What are Mukesh Ambani's relationships?"

```
execute -n aaaaaa5nb5t5cgg7nrintzf56dpmb3otezf3hycr47mbwocei5zwm5h4gm -m get_relationships -i '{"entity_id":"ENT-REL-001"}'
```

get_connected_entities

Natural Language: "Find all entities within 2 hops of Reliance CFO"

```
execute -n aaaaaa5nb5t5cgg7nrintzf56dpmb3otezf3hycr47mbwocei5zwm5h4gm -m get_connected_entities -i '{"entity_id":"ENT-REL-006","max_
```

check_insider_status

Natural Language: "Is Mukesh Ambani an insider of RELIANCE?"

```
execute -n aaaaaa5nb5t5cgg7nrintzf56dpmb3otezf3hycr47mbwocei5zwm5h4gm -m check_insider_status -i '{"entity_id":"ENT-REL-001","compar
```

Natural Language: "Is suspect SUS-001 connected to RELIANCE?"

```
execute -n aaaaaa5nb5t5cgg7nrintzf56dpmb3otezf3hycr47mbwocei5zwm5h4gm -m check_insider_status -i '{"entity_id":"SUS-001","company_sy
```

get_company_insiders

Natural Language: "List all insiders of RELIANCE"

```
execute -n aaaaaa5nb5t5cgg7nrintzf56dpmb3otezf3hycr47mbwocei5zwm5h4gm -m get_company_insiders -i '{"company_symbol":"RELIANCE"}'
```

Natural Language (Fuzzy): "Get INFY insiders"

```
execute -n aaaaaa5nb5t5cgg7nrintzf56dpmb3otezf3hycr47mbwocei5zwm5h4gm -m get_company_insiders -i '{"company_symbol":"INF"}'
```

are_entities_connected

Natural Language: "Is suspect SUS-001 connected to Reliance CFO?"

```
execute -n aaaaaa5nb5t5cgg7nrintzf56dpmb3otezf3hycr47mbwocei5zwm5h4gm -m are_entities_connected -i '{"entity_id_1":"SUS-001","entity
```

get_family_members

Natural Language: "Who are Mukesh Ambani's family members?"

```
execute -n aaaaaa5nb5t5cgg7nrintzf56dpmb3otezf3hycr47mbwocei5zwm5h4gm -m get_family_members -i '{"entity_id":"ENT-REL-001"}'
```

tools

Natural Language: "What methods are available in entity relationship?"

```
execute -n aaaaaa5nb5t5cgg7nrintzf56dpmb3otezf3hycr47mbwocei5zwm5h4gm -m tools
```

3. Trade Data MCP

Contract ID: `aaaaaa7bmmtxibdkvxut2rpyuplk57gadst7qlz5hkl7ugalw6co4lsryy`

get_context

Natural Language: "Show me recently queried symbols and accounts"

```
execute -n aaaaaa7bmmtxibdkvxut2rpyuplk57gadst7qlz5hkl7ugalw6co4lsryy -m get_context
```

get_trade

Natural Language: "Get trade details for trade ID IBM_1737225600000_ACC017"

```
execute -n aaaaaa7bmmtxibdkvxut2rpyuplk57gadst7qlz5hkl7ugalw6co4lsryy -m get_trade -i '{"trade_id":"IBM_1737225600000_ACC017"}'
```

get_trades_by_symbol

Natural Language: "Get the last 10 trades for IBM"

```
execute -n aaaaaa7bmmtxibdkvxut2rpyuplk57gadst7qlz5hkl7ugalw6co4lsryy -m get_trades_by_symbol -i '{"symbol":"IBM","limit":10}'
```

Natural Language (Fuzzy): "Show me Apple trades"

```
execute -n aaaaaa7bmmtxibdkvxut2rpyuplk57gadst7qlz5hkl7ugalw6co4lsryy -m get_trades_by_symbol -i '{"symbol":"AAP","limit":5}'
```

get_trades_by_account

Natural Language: "Get all trades by account ACC017"

```
execute -n aaaaaa7bmmtxibdkvxut2rpyuplk57gadst7qlz5hkl7ugalw6co4lsryy -m get_trades_by_account -i '{"account_id":"ACC017","limit":1
```

get_trades_by_accounts

Natural Language: "Get IBM trades for accounts ACC001, ACC002, ACC003"

```
execute -n aaaaaa7bmmtxibdkvxut2rpyuplk57gadst7qlz5hkl7ugalw6co4lsryy -m get_trades_by_accounts -i '{"account_ids":"ACC001,ACC002,A
```

analyze_volume

Natural Language: "Analyze trading volume for AAPL"

```
execute -n aaaaaa7bmmtxibdkvxut2rpyuplk57gadst7qlz5hkl7ugalw6co4lsryy -m analyze_volume -i '{"symbol":"AAPL"}'
```

detect_volume_anomaly

Natural Language: "Are there any volume anomalies on MSFT?"

```
execute -n aaaaaa7bmmtxibdkvxut2rpyuplk57gadst7qlz5hkl7ugalw6co4lsryy -m detect_volume_anomaly -i '{"symbol":"MSFT"}'
```

get_top_traders

Natural Language: "Who are the top 5 traders on GOOGL?"

```
execute -n aaaaaa7bmmtxibdkvxut2rpyuplk57gadst7qlz5hkl7ugalw6co4lsryy -m get_top_traders -i '{"symbol":"GOOGL","limit":5}'
```

get_large_orders

Natural Language: "Find large orders over $1,000,000"

```
execute -n aaaaaa7bmmtxibdkvxut2rpyuplk57gadst7qlz5hkl7ugalw6co4lsryy -m get_large_orders -i '{"min_value":1000000}'
```

get_account_profile

Natural Language: "Show me the trading profile for account ACC042"

```
execute -n aaaaaa7bmmtxibdkvxut2rpyuplk57gadst7qlz5hkl7ugalw6co4lsryy -m get_account_profile -i '{"account_id":"ACC042"}'
```

tools

Natural Language: "What methods are available in trade data?"

```
execute -n aaaaaa7bmmtxibdkvxut2rpyuplk57gadst7qlz5hkl7ugalw6co4lsryy -m tools
```

4. UPSI Database MCP

Contract ID: `aaaaaaytejhz4sy6rhqobwg2sqvloy5276ouvo6ejod2fzt4rtobggxtf4`

get_context

Natural Language: "Show me recent UPSI queries"

```
execute -n aaaaaaytejhz4sy6rhqobwg2sqvloy5276ouvo6ejod2fzt4rtobggxtf4 -m get_context
```

get_upsi

Natural Language: "Get UPSI record UPSI-001"

```
execute -n aaaaaaytejhz4sy6rhqobwg2sqvloy5276ouvo6ejod2fzt4rtobggxtf4 -m get_upsi -i '{"upsi_id":"UPSI-001"}'
```

Natural Language (Fuzzy): "Get UPSI for 001"

```
execute -n aaaaaaytejhz4sy6rhqobwg2sqvloy5276ouvo6ejod2fzt4rtobggxtf4 -m get_upsi -i '{"upsi_id":"001"}'
```

get_active_upsi

Natural Language: "What active UPSI exists for RELIANCE?"

```
execute -n aaaaaaytejhz4sy6rhqobwg2sqvloy5276ouvo6ejod2fzt4rtobggxtf4 -m get_active_upsi -i '{"company_symbol":"RELIANCE"}'
```

Natural Language: "Show active UPSI for Infosys"

```
execute -n aaaaaaytejhz4sy6rhqobwg2sqvloy5276ouvo6ejod2fzt4rtobggxtf4 -m get_active_upsi -i '{"company_symbol":"INFY"}'
```

get_upsi_access_log

Natural Language: "Who accessed UPSI-001 in the last month?"

```
execute -n aaaaaaytejhz4sy6rhqobwg2sqvloy5276ouvo6ejod2fzt4rtobggxtf4 -m get_upsi_access_log -i '{"upsi_id":"UPSI-001","from_timesta
```

get_access_by_person

Natural Language: "What UPSI did Mukesh Ambani access in the last 30 days?"

```
execute -n aaaaaaytejhz4sy6rhqobwg2sqvloy5276ouvo6ejod2fzt4rtobggxtf4 -m get_access_by_person -i '{"accessor_entity_id":"ENT-REL-001
```

Natural Language: "What UPSI did Reliance CFO access?"

```
execute -n aaaaaaytejhz4sy6rhqobwg2sqvloy5276ouvo6ejod2fzt4rtobggxtf4 -m get_access_by_person -i '{"accessor_entity_id":"ENT-REL-006
```

check_upsi_access_before

Natural Language: "Did suspect SUS-001 access RELIANCE UPSI before trading?"

```
execute -n aaaaaaytejhz4sy6rhqobwg2sqvloy5276ouvo6ejod2fzt4rtobggxtf4 -m check_upsi_access_before -i '{"entity_id":"SUS-001","compar
```

get_trading_window

Natural Language: "Is RELIANCE trading window open or closed?"

```
execute -n aaaaaaytejhz4sy6rhqobwg2sqvloy5276ouvo6ejod2fzt4rtobggxtf4 -m get_trading_window -i '{"company_symbol":"RELIANCE"}'
```

Natural Language: "Check TCS trading window status"

```
execute -n aaaaaaytejhz4sy6rhqobwg2sqvloy5276ouvo6ejod2fzt4rtobggxtf4 -m get_trading_window -i '{"company_symbol":"TCS"}'
```

check_window_violation

Natural Language: "Did Mukesh Ambani trade RELIANCE during a closed window?"

```
execute -n aaaaaaytejhz4sy6rhqobwg2sqvloy5276ouvo6ejod2fzt4rtobggxtf4 -m check_window_violation -i '{"entity_id":"ENT-REL-001","comp
```

get_upsi_accessors

Natural Language: "Who has accessed UPSI-001?"

```
execute -n aaaaaaytejhz4sy6rhqobwg2sqvloy5276ouvo6ejod2fzt4rtobggxtf4 -m get_upsi_accessors -i '{"upsi_id":"UPSI-001"}'
```

tools

Natural Language: "What methods are available in UPSI database?"

```
execute -n aaaaaaytejhz4sy6rhqobwg2sqvloy5276ouvo6ejod2fzt4rtobggxtf4 -m tools
```

---

5. Jira MCP

Contract ID: `aaaaaa22qx66rbpoh6ak2mmjfb5kc2ryjnoqzp33ujzi73u3aze5wkij7e`

create_ticket

Natural Language: "Create a high priority Jira ticket for investigating suspicious trading"

```
execute -n aaaaaa22qx66rbpoh6ak2mmjfb5kc2ryjnoqzp33ujzi73u3aze5wkij7e -m create_ticket -i '{"summary":"Suspicious trading detected f
```

create_case_ticket

Natural Language: "Create a surveillance case ticket for case CASE-001 investigating Mukesh Ambani"

```
execute -n aaaaaa22qx66rbpoh6ak2mmjfb5kc2ryjnoqzp33ujzi73u3aze5wkij7e -m create_case_ticket -i '{"case_id":"CASE-001","subject_entit
```

close_ticket

Natural Language: "Close ticket SURV-123 as Done"

```
execute -n aaaaaa22qx66rbpoh6ak2mmjfb5kc2ryjnoqzp33ujzi73u3aze5wkij7e -m close_ticket -i '{"ticket_key":"SURV-123","resolution":"Don
```

get_ticket

Natural Language: "Get details for Jira ticket SURV-123"

```
execute -n aaaaaa22qx66rbpoh6ak2mmjfb5kc2ryjnoqzp33ujzi73u3aze5wkij7e -m get_ticket -i '{"ticket_key":"SURV-123"}'
```

add_comment

Natural Language: "Add a comment to ticket SURV-123 about the investigation progress"

```
execute -n aaaaaa22qx66rbpoh6ak2mmjfb5kc2ryjnoqzp33ujzi73u3aze5wkij7e -m add_comment -i '{"ticket_key":"SURV-123","comment":"Investi
```

update_ticket_status

Natural Language: "Move ticket SURV-123 to In Progress"

```
execute -n aaaaaa22qx66rbpoh6ak2mmjfb5kc2ryjnoqzp33ujzi73u3aze5wkij7e -m update_ticket_status -i '{"ticket_key":"SURV-123","new_stat
```

tools

Natural Language: "What methods are available in Jira?"

```
execute -n aaaaaa22qx66rbpoh6ak2mmjfb5kc2ryjnoqzp33ujzi73u3aze5wkij7e -m tools
```

Key Entity IDs Reference

Entity ID	Name	Type
ENT-REL-001	Mukesh Ambani	Reliance CMD
ENT-REL-006	Srikanth Venkatachari	Reliance CFO
ENT-INF-001	Salil Parekh	Infosys CEO
ENT-TCS-001	K. Krithivasan	TCS CEO
SUS-001	Vikram Shell Holdings	Suspect
SUS-003	Deepak Malhotra	Suspect
BRK-001	Alpha Securities	Broker
ACC017	Trading Account 17	Account
ACC042	Trading Account 42	Account

Company Symbols

- `RELIANCE` - Reliance Industries
- `INFY` - Infosys
- `TCS` - Tata Consultancy Services
- `HDFCBANK` - HDFC Bank
- `IBM` - IBM (US Stock)
- `AAPL` - Apple (US Stock)
- `MSFT` - Microsoft (US Stock)
- `GOOGL` - Google (US Stock)
- `TSLA` - Tesla (US Stock)