Koen Decorte

CD INVEST
BI · AI · IBMi

The Zero-Code Revolution

Natural Language Processing Takes Over IBM i

www.cdinvest.eu

Who are we ?

CD INVEST
BI · AI · IBMi

International IBM i ISV and
IBM business partner .

located in Antwerp, Belgium

Expertise in RPG, SQL,
Python, HTML, Ai,
NodeJS, Linux...

Working with IBM i
and its predecessors
for more than 40
year

IBM Champion
since 2018 and
CEAC member

Applications : CDErp,
CDReport CD-Account
accountancy, CDVts
and MES.

What others talk about,
we do.

Agenda

- The current state of IBM i programming
- Introduction to NLP and its evolution
- The convergence of NLP and IBM i
- Practical applications and demonstrations
- Implementation roadmap
- Challenges and considerations
- Future outlook

The current state of IBM i programming

The Promise of Zero-Code

- Democratization of system management
- Reduced technical barriers
- Faster development cycles
- Focus on business problems rather than syntax

Poll the Audience

- What programming languages do you currently use?

- Have you experimented with any NLP tools?

- What's your biggest challenge in IBM i development?

The IBM i Legacy

- Brief history of IBM i (AS/400, iSeries, System i)
- Core strengths of the platform
- Evolution of programming approaches

Current Programming Landscape

- RPG, COBOL, CL, SQL
- Modern languages on IBM i (Java, Python, PHP, Node.js)
- Integrated development environments
- Current automation tools

The Paradox of IBM i

- Robust and reliable architecture
- Complex programming requirements
- Skills gap challenges
- Need for modernization while preserving investments

The Vision

- Conversational interaction with IBM i systems
- Natural language queries replacing SQL
- Voice-driven system management
- AI assistants collaborating with human operators

What We'll Cover Today

- Technical underpinnings of NLP on IBM i
- Practical demonstrations
- Implementation approach
- Ethical considerations

CD INVEST
BI · AI · IBMi

Understanding NLP fundamentals

What is Natural Language Processing?

- Definition and core concepts
- Historical evolution
- Key components: tokenization, parsing, semantics
- Relationship to AI and machine learning

The NLP Revolution Timeline

- Early rule-based systems (1950s-1980s)
- Statistical NLP (1990s-2010s)
- Neural networks and deep learning (2010s)
- Transformer models (2017-present)
- Large language models (2020-present)

How Modern NLP Works

- Word embeddings and vector spaces
- Attention mechanisms
- Transformer architectures
- Context understanding
- Fine-tuning approaches

Key NLP Capabilities

- Sentiment analysis
- Named entity recognition
- Syntax parsing
- Question answering
- Text generation
- Translation
- Summarization

Large Language Models

- GPT, Gemma, DeepSeek, and other foundation models
- Training methodologies
- Parameter counts and model sizes
- Capabilities and limitations

Enterprise NLP Applications

- Document processing
- Customer service automation
- Knowledge management
- Code generation
- Business intelligence

NLP for Technical Systems

- Code generation capabilities
- System monitoring through natural language
- Automated documentation
- Conversational interfaces to databases

Code Generation Examples

- SQL generation from natural language
- API development through conversation
- Database schema design through description
- RPG code translation to RPG free or other languages

Understanding NLP Limitations

- Hallucinations and confidence issues
- Context window constraints
- Domain-specific knowledge challenges
- The need for human oversight

The Path from NLP to Zero-Code

- Supervised learning on existing codebases
- Domain-specific language models
- Integration with traditional systems
- Hybrid human-AI development approaches

CD INVEST
BI · AI · IBMi

IBM i meets NLP

IBM i Architecture Overview

- Integrated DB2 database
- Single-level storage
- Object-based security
- Subsystem architecture
- Technology Independent Machine Interface (TIMI)

Why IBM i is Uniquely Positioned for NLP

- Metadata-rich environment
- Integrated database
- Object-oriented architecture
- Structured system APIs
- Historical emphasis on abstract interfaces

The NLP Integration Points

- Command line interfaces
- Database access layers
- System APIs
- IFS (Integrated File System)
- IBM i Navigator

Architectural Components

- NLP processing engine (local vs. cloud)
- Intent recognition system
- IBM i connector framework
- Security and authentication layer
- Audit and validation components

NLP-Driven System Operations

- Starting/stopping subsystems
- Managing job queues
- Performance tuning
- Security administration
- Backup and recovery operations

Natural Language Database Interaction

- Conversational data retrieval
- Schema exploration through dialog
- Data visualization requests
- Complex query generation
- Data modification safeguards

Application Development Transformation

- Describing business rules in natural language
- Automatic code generation
- Testing through conversation
- Documentation generation
- Continuous improvement through feedback

Integration with Existing Systems

- Mapping NLP intents to CL commands
- Bridging to RPG applications
- SQL generation for complex queries
- API creation through conversation
- Legacy modernization approaches

Zero-Code Development Process

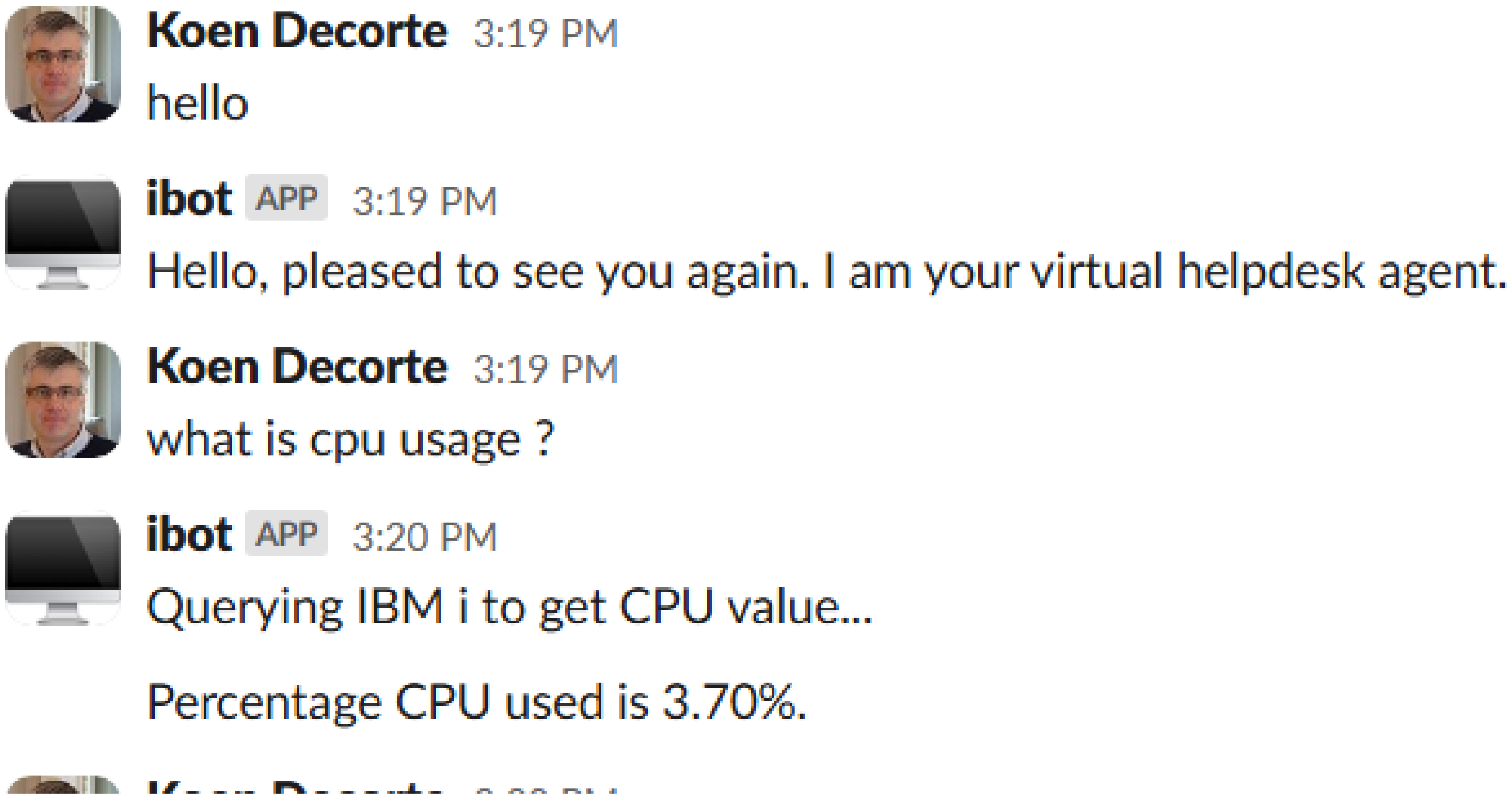
- Business requirement articulation
- NLP interpretation and clarification
- Solution design through dialog
- Implementation without manual coding
- Testing and refinement conversations

Practical applications and demonstrations

Demo : System Inquiry

- "Show me active jobs consuming the most CPU"
- "Which users are currently logged in?"
- “Reset a user Password"
- "How much free storage remains in ASP1?"

Query cpu usage



Koen Decorte 3:19 PM
hello

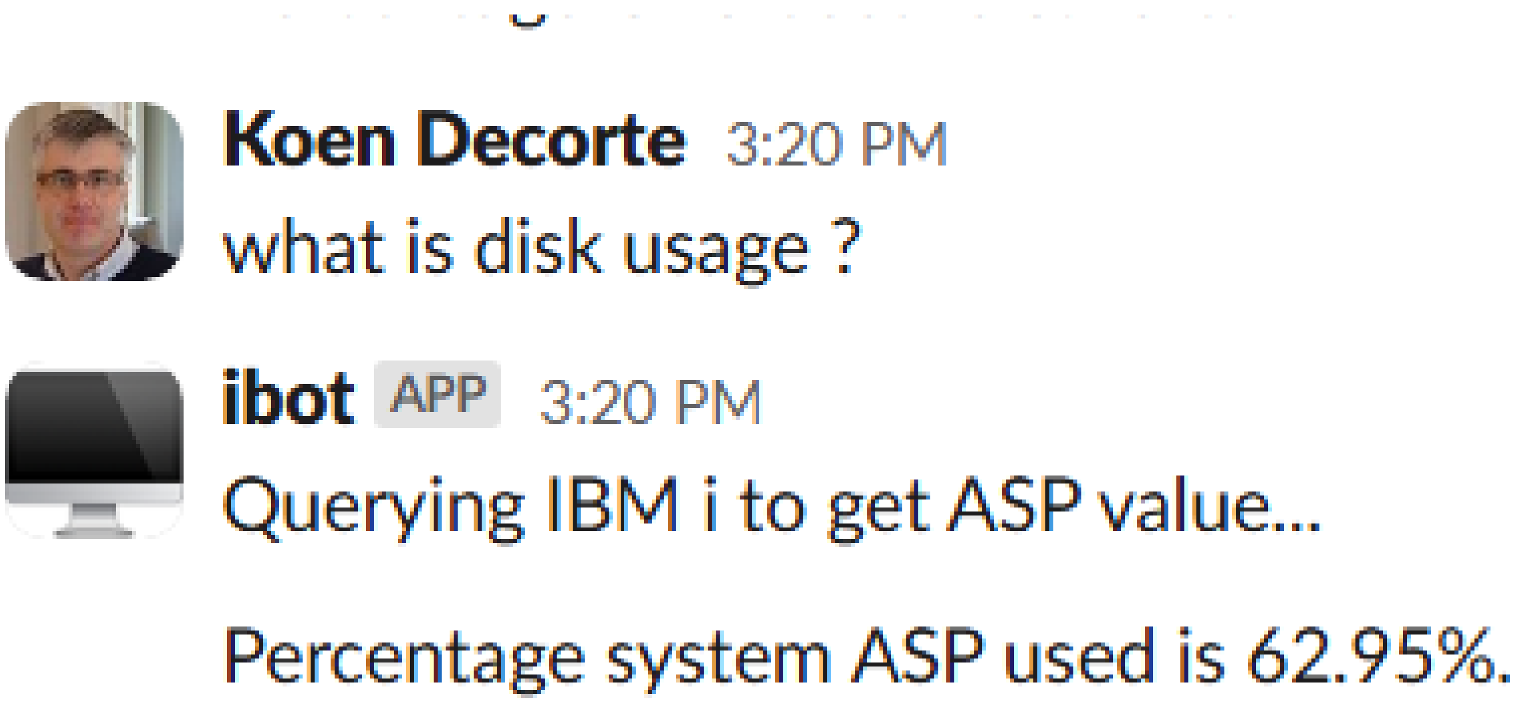
ibot APP 3:19 PM
Hello, pleased to see you again. I am your virtual helpdesk agent.

Koen Decorte 3:19 PM
what is cpu usage ?

ibot APP 3:20 PM
Querying IBM i to get CPU value...

Percentage CPU used is 3.70%.

Query disk usage



Koen Decorte 3:20 PM

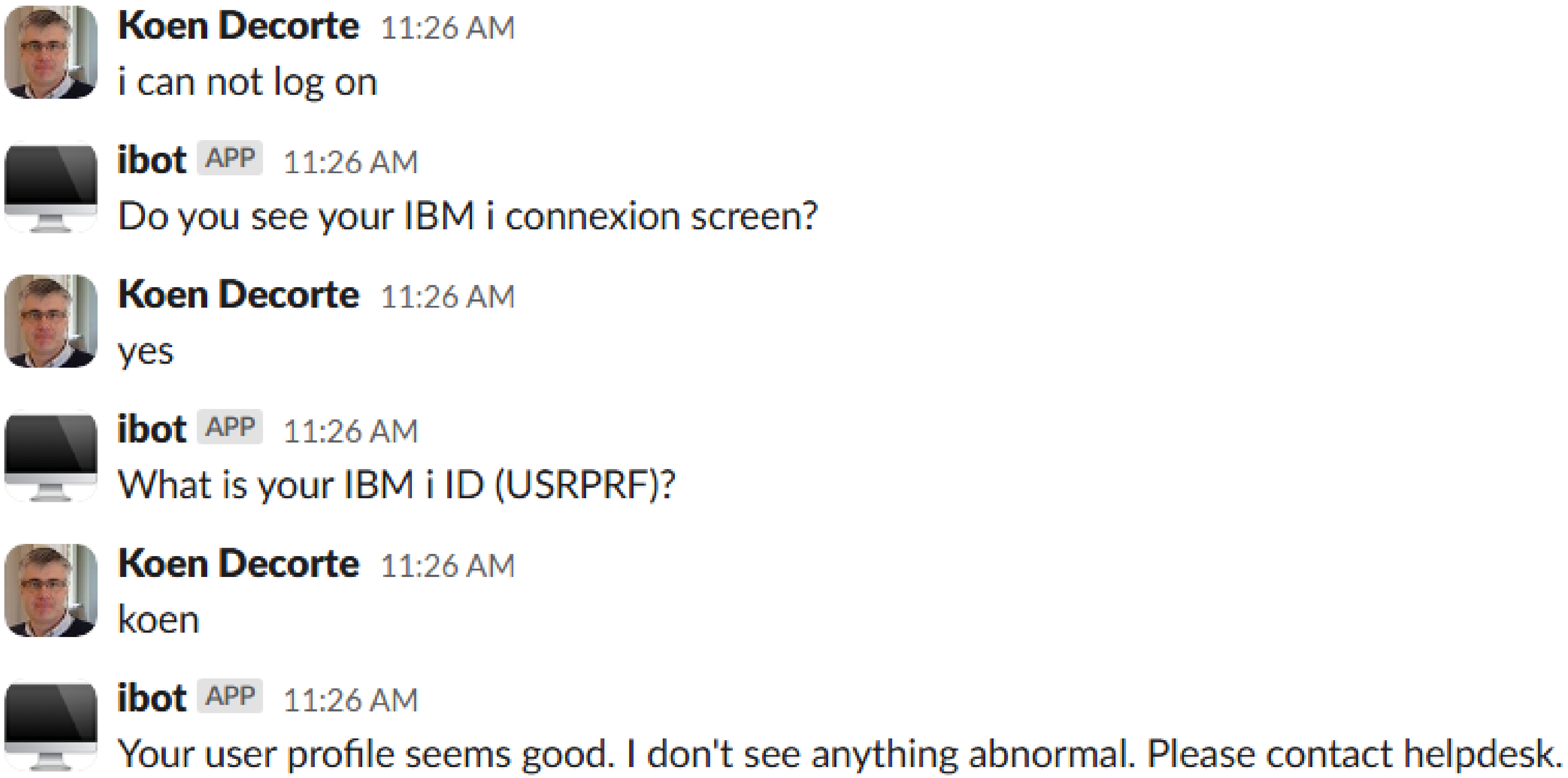
what is disk usage ?

ibot APP 3:20 PM

Querying IBM i to get ASP value...

Percentage system ASP used is 62.95%.

Reset user password



Koen Decorte 11:26 AM
i can not log on

ibot APP 11:26 AM
Do you see your IBM i connexion screen?

Koen Decorte 11:26 AM
yes

ibot APP 11:26 AM
What is your IBM i ID (USRPRF)?

Koen Decorte 11:26 AM
koen

ibot APP 11:26 AM
Your user profile seems good. I don't see anything abnormal. Please contact helpdesk.

Demo : Application Modification

- "Add a discount field to the customer master"
- "Create a validation rule for the new discount field"
- "Modify the order entry screen to display the discount"
- "Update all reports to include discount information"

Demo : Modernization

- "Convert this RPG III program to free-form RPG"
- "Create a REST API for the inventory system"
- "Generate a web interface for customer maintenance"
- "Implement mobile access to sales data"

DEMO : DevOps

- Check code on Quality
- Check code security rules
- Generate code documentation

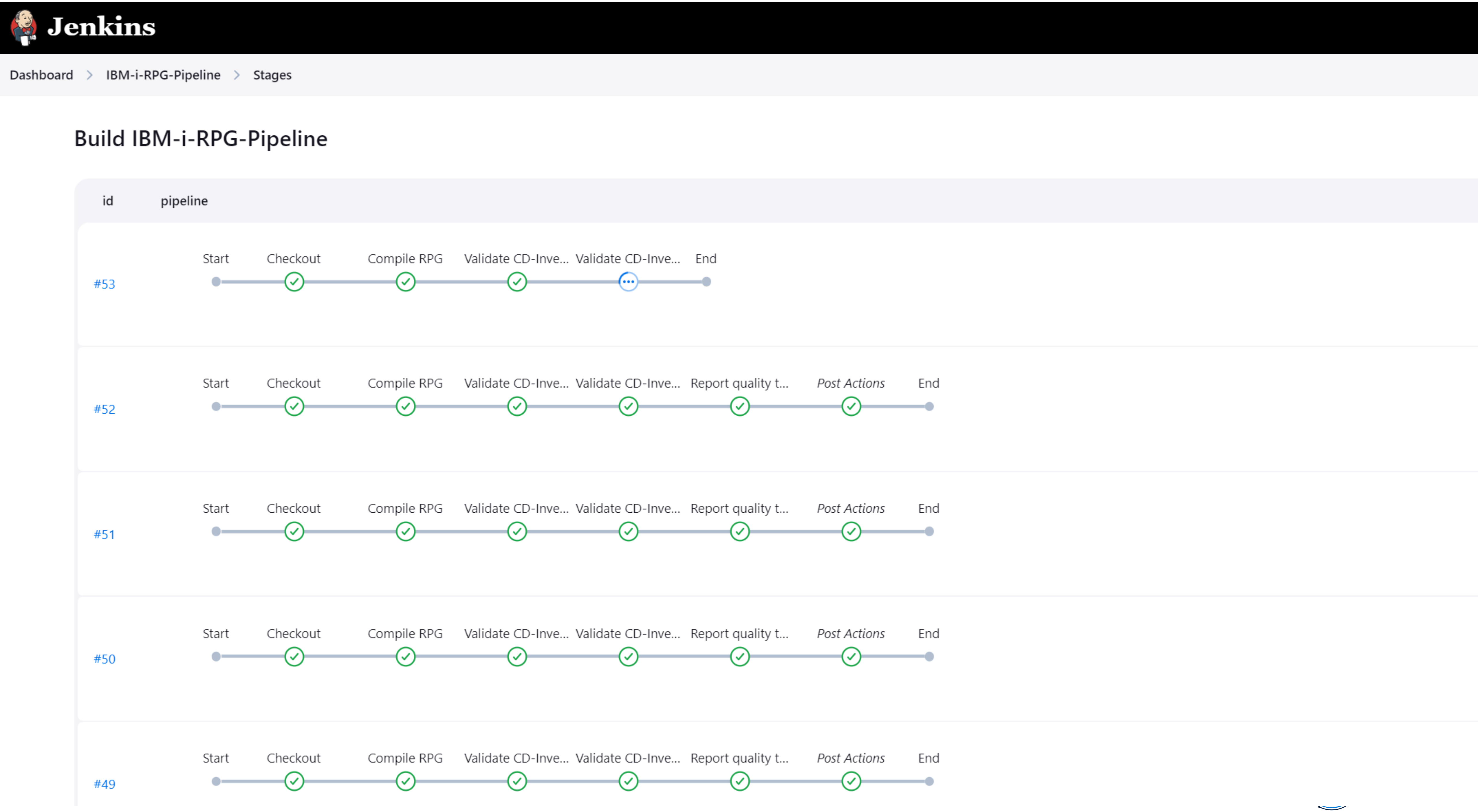
Security Analysis with AI

> Dedicated security scanning identifies potential vulnerabilities in RPG code before deployment.

Security Vulnerabilities Detected:

- SQL injection vulnerabilities
- Improper input validation
- Hardcoded credentials
- Insecure authorization
- Buffer overflows
- Path manipulation issues
- Missing error handling in security-critical code

```
{
  "secure": false,
  "vulnerabilities": [
    {
      "line": 125,
      "severity": "CRITICAL",
      "vulnerability_type": "SQL_INJECTION",
      "message": "Direct user input in SQL query",
      "remediation": "Use prepared statement",
      "cwe_id": "CWE-89"
    }
  ]
}
```

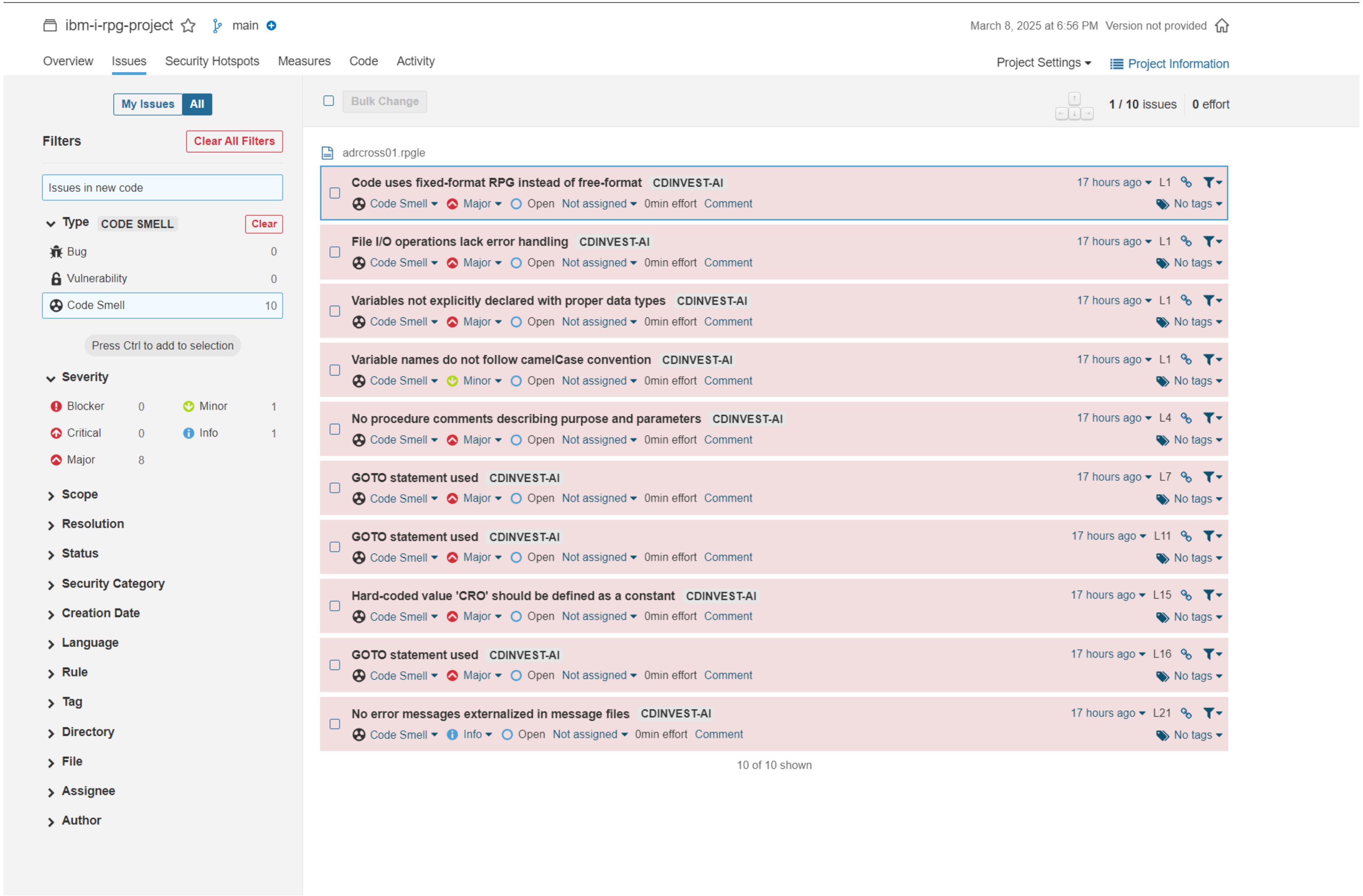


Jenkins

Dashboard > IBM-i-RPG-Pipeline > Stages

Build IBM-i-RPG-Pipeline

id	pipeline
#53	Start → Checkout ✓ → Compile RPG ✓ → Validate CD-Inve... ✓ → Validate CD-Inve... ⋯ → End
#52	Start → Checkout ✓ → Compile RPG ✓ → Validate CD-Inve... ✓ → Validate CD-Inve... ✓ → Report quality t... ✓ → *Post Actions* ✓ → End
#51	Start → Checkout ✓ → Compile RPG ✓ → Validate CD-Inve... ✓ → Validate CD-Inve... ✓ → Report quality t... ✓ → *Post Actions* ✓ → End
#50	Start → Checkout ✓ → Compile RPG ✓ → Validate CD-Inve... ✓ → Validate CD-Inve... ✓ → Report quality t... ✓ → *Post Actions* ✓ → End
#49	Start → Checkout ✓ → Compile RPG ✓ → Validate CD-Inve... ✓ → Validate CD-Inve... ✓ → Report quality t... ✓ → *Post Actions* ✓ → End



adrcross01.rpgle

☐ Code uses fixed-format RPG instead of free-format CDINVEST-AI — 17 hours ago L1
Code Smell · Major · Open · Not assigned · 0min effort · Comment · No tags

☐ File I/O operations lack error handling CDINVEST-AI — 17 hours ago L1
Code Smell · Major · Open · Not assigned · 0min effort · Comment · No tags

☐ Variables not explicitly declared with proper data types CDINVEST-AI — 17 hours ago L1
Code Smell · Major · Open · Not assigned · 0min effort · Comment · No tags

☐ Variable names do not follow camelCase convention CDINVEST-AI — 17 hours ago L1
Code Smell · Minor · Open · Not assigned · 0min effort · Comment · No tags

☐ No procedure comments describing purpose and parameters CDINVEST-AI — 17 hours ago L4
Code Smell · Major · Open · Not assigned · 0min effort · Comment · No tags

☐ GOTO statement used CDINVEST-AI — 17 hours ago L7
Code Smell · Major · Open · Not assigned · 0min effort · Comment · No tags

☐ GOTO statement used CDINVEST-AI — 17 hours ago L11
Code Smell · Major · Open · Not assigned · 0min effort · Comment · No tags

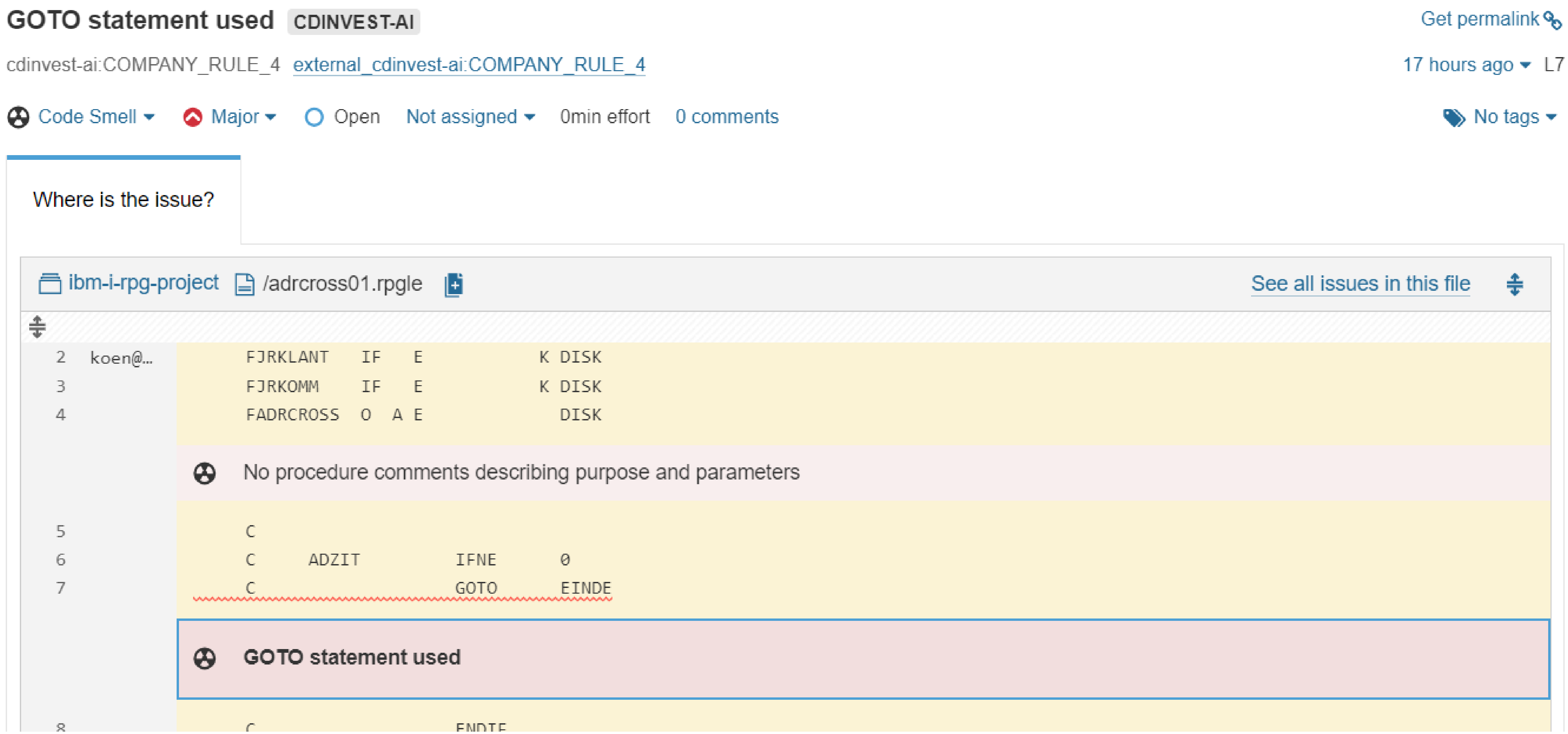
☐ Hard-coded value 'CRO' should be defined as a constant CDINVEST-AI — 17 hours ago L15
Code Smell · Major · Open · Not assigned · 0min effort · Comment · No tags

☐ GOTO statement used CDINVEST-AI — 17 hours ago L16
Code Smell · Major · Open · Not assigned · 0min effort · Comment · No tags

☐ No error messages externalized in message files CDINVEST-AI — 17 hours ago L21
Code Smell · Info · Open · Not assigned · 0min effort · Comment · No tags

10 of 10 shown

CD INVEST
BI · AI · IBMi



GOTO statement used CDINVEST-AI

Get permalink

cdinvest-ai:COMPANY_RULE_4 external_cdinvest-ai:COMPANY_RULE_4

17 hours ago ▾ L7

Code Smell ▾ Major ▾ ○ Open Not assigned ▾ 0min effort 0 comments

No tags ▾

Where is the issue?

ibm-i-rpg-project /adrcross01.rpgle

See all issues in this file

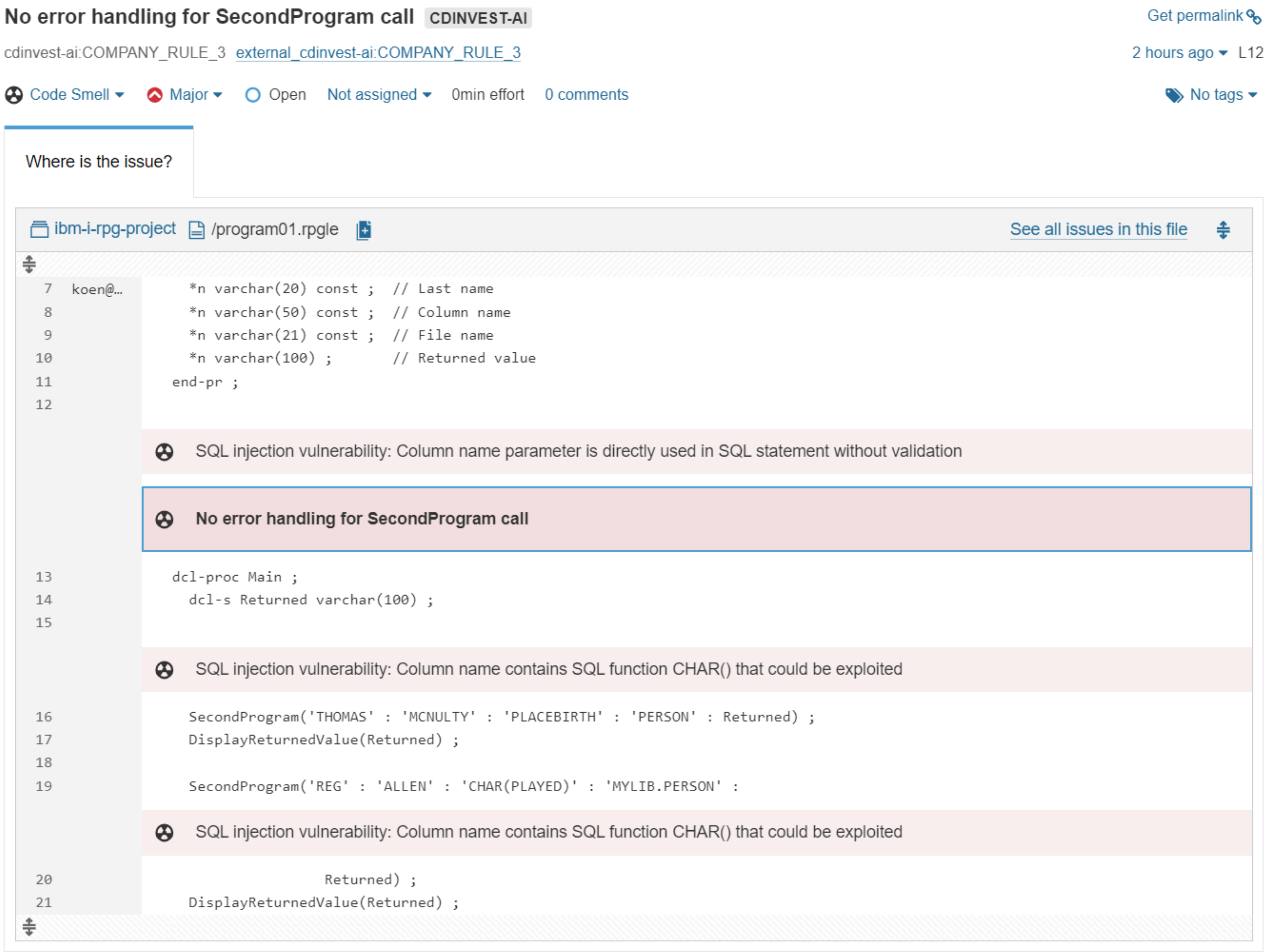
```
2  koen@…     FJRKLANT   IF   E           K DISK
3             FJRKOMM    IF   E           K DISK
4             FADRCROSS  O   A E             DISK
```

No procedure comments describing purpose and parameters

```
5             C
6             C     ADZIT         IFNE      0
7             C                   GOTO      EINDE
```

GOTO statement used

```
8             C                   ENDIF
```



No error handling for SecondProgram call CDINVEST-AI

cdinvest-ai:COMPANY_RULE_3 external_cdinvest-ai:COMPANY_RULE_3

2 hours ago ▾ L12

```
 7  koen@...     *n varchar(20) const ;  // Last name
 8               *n varchar(50) const ;  // Column name
 9               *n varchar(21) const ;  // File name
10               *n varchar(100) ;       // Returned value
11             end-pr ;
12
```

SQL injection vulnerability: Column name parameter is directly used in SQL statement without validation

No error handling for SecondProgram call

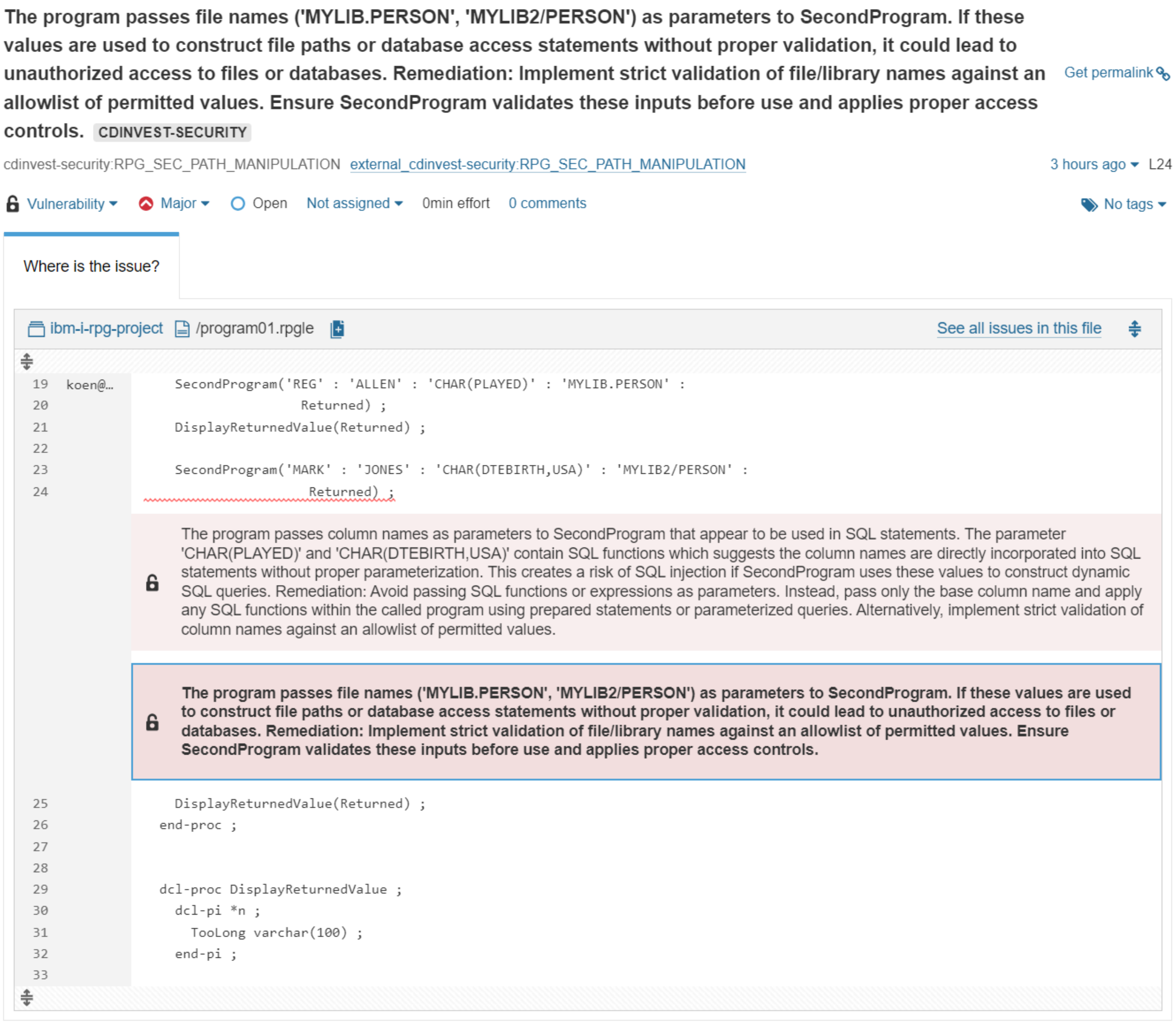
```
13             dcl-proc Main ;
14               dcl-s Returned varchar(100) ;
15
```

SQL injection vulnerability: Column name contains SQL function CHAR() that could be exploited

```
16               SecondProgram('THOMAS' : 'MCNULTY' : 'PLACEBIRTH' : 'PERSON' : Returned) ;
17               DisplayReturnedValue(Returned) ;
18
19               SecondProgram('REG' : 'ALLEN' : 'CHAR(PLAYED)' : 'MYLIB.PERSON' :
```

SQL injection vulnerability: Column name contains SQL function CHAR() that could be exploited

```
20                                Returned) ;
21               DisplayReturnedValue(Returned) ;
```



The program passes file names ('MYLIB.PERSON', 'MYLIB2/PERSON') as parameters to SecondProgram. If these values are used to construct file paths or database access statements without proper validation, it could lead to unauthorized access to files or databases. Remediation: Implement strict validation of file/library names against an allowlist of permitted values. Ensure SecondProgram validates these inputs before use and applies proper access controls. CDINVEST-SECURITY

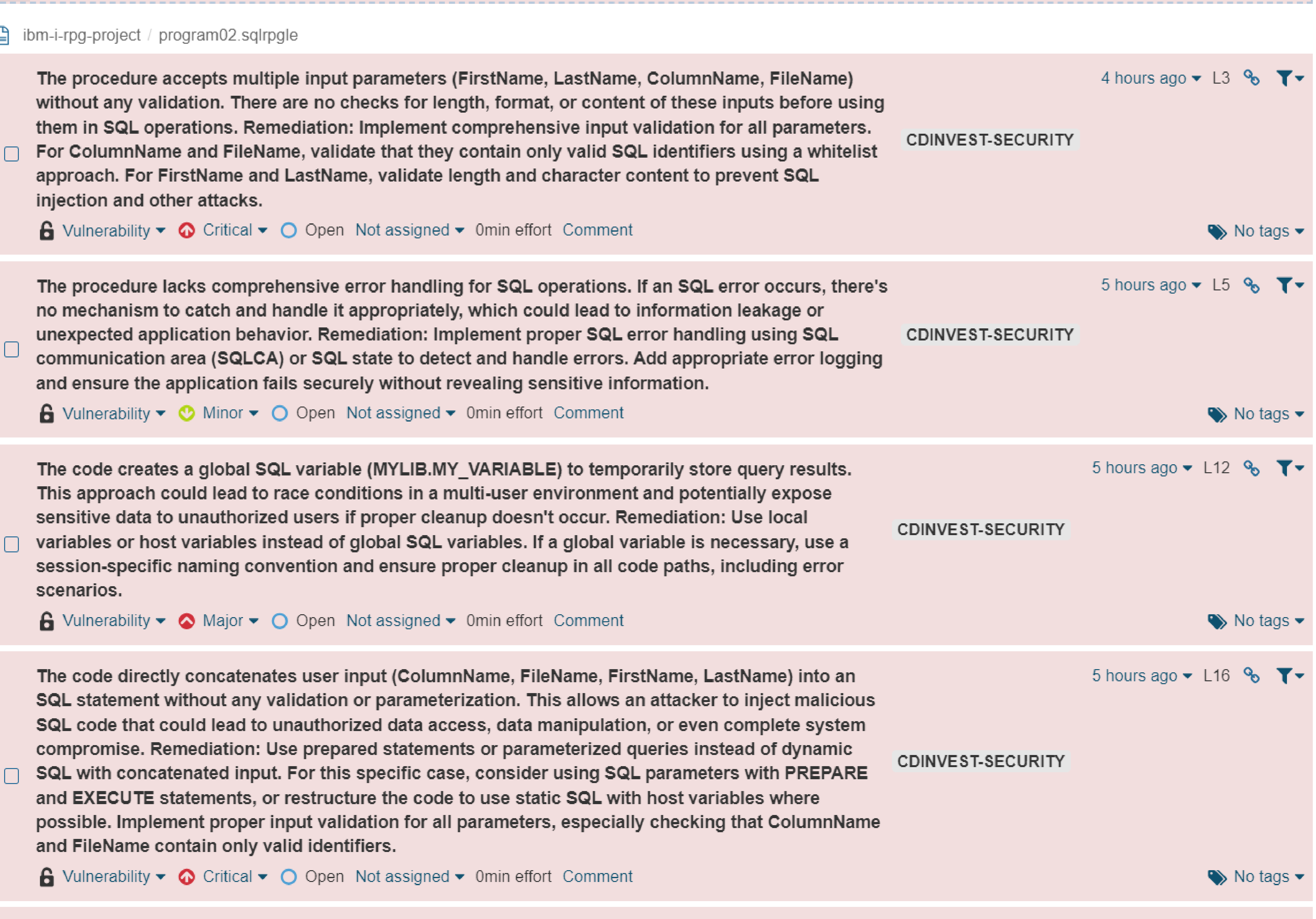
cdinvest-security:RPG_SEC_PATH_MANIPULATION external_cdinvest-security:RPG_SEC_PATH_MANIPULATION

```
19  koen@…    SecondProgram('REG' : 'ALLEN' : 'CHAR(PLAYED)' : 'MYLIB.PERSON' :
20                             Returned) ;
21            DisplayReturnedValue(Returned) ;
22
23            SecondProgram('MARK' : 'JONES' : 'CHAR(DTEBIRTH,USA)' : 'MYLIB2/PERSON' :
24                              Returned) ;
```

The program passes column names as parameters to SecondProgram that appear to be used in SQL statements. The parameter 'CHAR(PLAYED)' and 'CHAR(DTEBIRTH,USA)' contain SQL functions which suggests the column names are directly incorporated into SQL statements without proper parameterization. This creates a risk of SQL injection if SecondProgram uses these values to construct dynamic SQL queries. Remediation: Avoid passing SQL functions or expressions as parameters. Instead, pass only the base column name and apply any SQL functions within the called program using prepared statements or parameterized queries. Alternatively, implement strict validation of column names against an allowlist of permitted values.

The program passes file names ('MYLIB.PERSON', 'MYLIB2/PERSON') as parameters to SecondProgram. If these values are used to construct file paths or database access statements without proper validation, it could lead to unauthorized access to files or databases. Remediation: Implement strict validation of file/library names against an allowlist of permitted values. Ensure SecondProgram validates these inputs before use and applies proper access controls.

```
25            DisplayReturnedValue(Returned) ;
26          end-proc ;
27
28
29          dcl-proc DisplayReturnedValue ;
30            dcl-pi *n ;
31              TooLong varchar(100) ;
32            end-pi ;
33
```



ibm-i-rpg-project / program02.sqlrpgle

☐ The procedure accepts multiple input parameters (FirstName, LastName, ColumnName, FileName) without any validation. There are no checks for length, format, or content of these inputs before using them in SQL operations. Remediation: Implement comprehensive input validation for all parameters. For ColumnName and FileName, validate that they contain only valid SQL identifiers using a whitelist approach. For FirstName and LastName, validate length and character content to prevent SQL injection and other attacks.

CDINVEST-SECURITY

4 hours ago L3

Vulnerability Critical Open Not assigned 0min effort Comment No tags

☐ The procedure lacks comprehensive error handling for SQL operations. If an SQL error occurs, there's no mechanism to catch and handle it appropriately, which could lead to information leakage or unexpected application behavior. Remediation: Implement proper SQL error handling using SQL communication area (SQLCA) or SQL state to detect and handle errors. Add appropriate error logging and ensure the application fails securely without revealing sensitive information.

CDINVEST-SECURITY

5 hours ago L5

Vulnerability Minor Open Not assigned 0min effort Comment No tags

☐ The code creates a global SQL variable (MYLIB.MY_VARIABLE) to temporarily store query results. This approach could lead to race conditions in a multi-user environment and potentially expose sensitive data to unauthorized users if proper cleanup doesn't occur. Remediation: Use local variables or host variables instead of global SQL variables. If a global variable is necessary, use a session-specific naming convention and ensure proper cleanup in all code paths, including error scenarios.

CDINVEST-SECURITY

5 hours ago L12

Vulnerability Major Open Not assigned 0min effort Comment No tags

☐ The code directly concatenates user input (ColumnName, FileName, FirstName, LastName) into an SQL statement without any validation or parameterization. This allows an attacker to inject malicious SQL code that could lead to unauthorized data access, data manipulation, or even complete system compromise. Remediation: Use prepared statements or parameterized queries instead of dynamic SQL with concatenated input. For this specific case, consider using SQL parameters with PREPARE and EXECUTE statements, or restructure the code to use static SQL with host variables where possible. Implement proper input validation for all parameters, especially checking that ColumnName and FileName contain only valid identifiers.

CDINVEST-SECURITY

5 hours ago L16

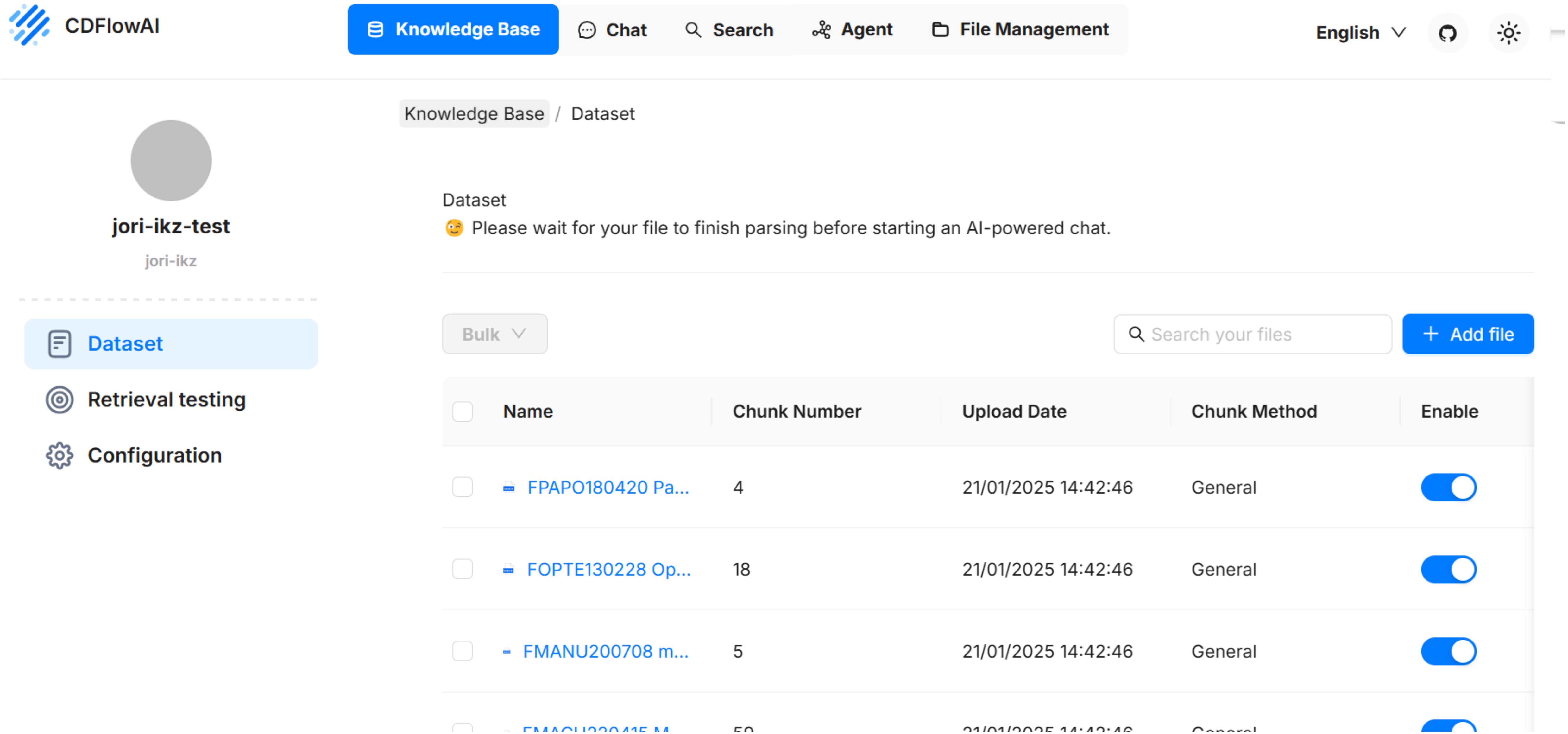
Vulnerability Critical Open Not assigned 0min effort Comment No tags

CD INVEST
BI · AI · IBMi

Demo : Database and document operations

"Show me customers who haven't ordered in 6 months"

- "Create a report of inventory items below reorder point"
- "What were our top-selling products last quarter?"
- "Find transactions with potential data quality issues"

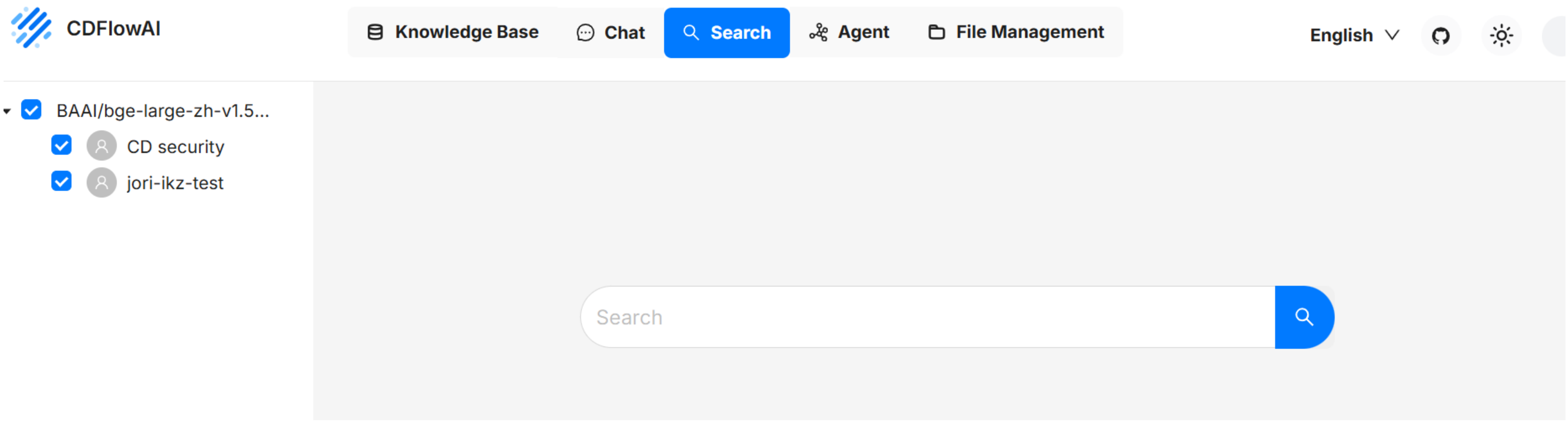


Knowledge Base / Dataset

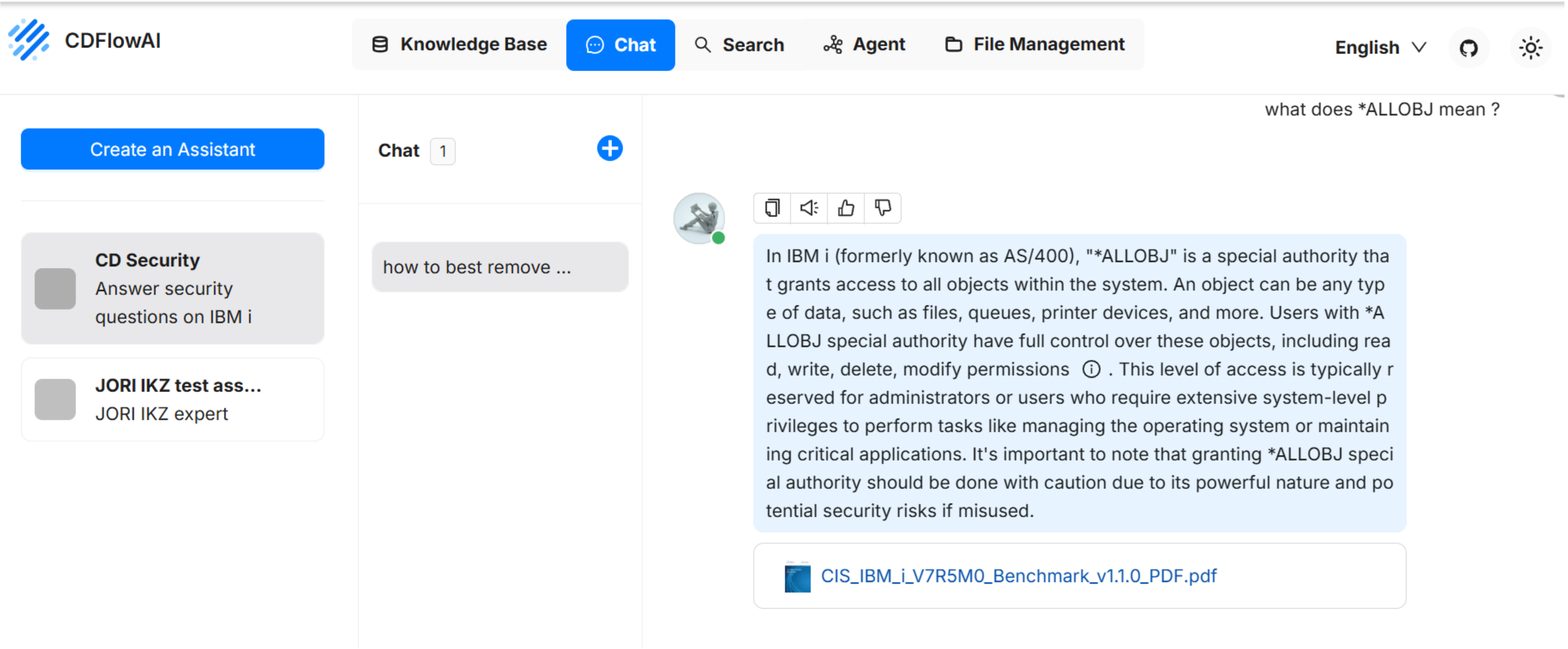
Dataset

Please wait for your file to finish parsing before starting an AI-powered chat.

☐	Name	Chunk Number	Upload Date	Chunk Method	Enable
☐	FPAPO180420 Pa...	4	21/01/2025 14:42:46	General	●
☐	FOPTE130228 Op...	18	21/01/2025 14:42:46	General	●
☐	FMANU200708 m...	5	21/01/2025 14:42:46	General	●



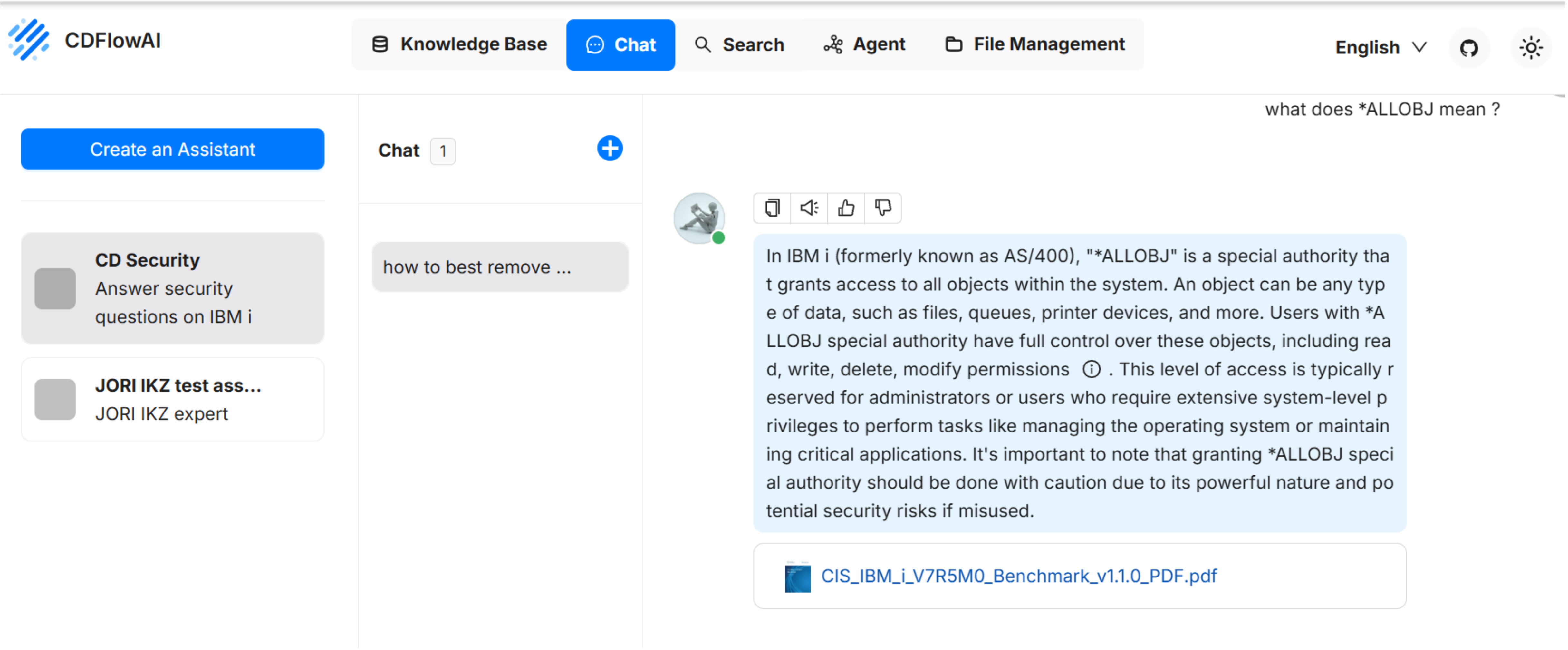
- ☑ BAAI/bge-large-zh-v1.5...
  - ☑ CD security
  - ☑ jori-ikz-test



what does *ALLOBJ mean ?

In IBM i (formerly known as AS/400), "*ALLOBJ" is a special authority that grants access to all objects within the system. An object can be any type of data, such as files, queues, printer devices, and more. Users with *ALLOBJ special authority have full control over these objects, including read, write, delete, modify permissions ⓘ . This level of access is typically reserved for administrators or users who require extensive system-level privileges to perform tasks like managing the operating system or maintaining critical applications. It's important to note that granting *ALLOBJ special authority should be done with caution due to its powerful nature and potential security risks if misused.

CIS_IBM_i_V7R5M0_Benchmark_v1.1.0_PDF.pdf



what does *ALLOBJ mean ?

In IBM i (formerly known as AS/400), "*ALLOBJ" is a special authority that grants access to all objects within the system. An object can be any type of data, such as files, queues, printer devices, and more. Users with *ALLOBJ special authority have full control over these objects, including read, write, delete, modify permissions ⓘ . This level of access is typically reserved for administrators or users who require extensive system-level privileges to perform tasks like managing the operating system or maintaining critical applications. It's important to note that granting *ALLOBJ special authority should be done with caution due to its powerful nature and potential security risks if misused.

CIS_IBM_i_V7R5M0_Benchmark_v1.1.0_PDF.pdf

CD INVEST
BI · AI · IBMi

Implementation roadmap

Readiness Assessment

- Technical infrastructure requirements
- Skills inventory
- Application portfolio analysis
- Business process documentation
- Data quality evaluation

Proof of Concept Framework

- Scope definition
- Success criteria
- Resource requirements
- Timeline expectations
- Risk mitigation strategies

Technical Architecture

- On-premises vs. cloud components
- Security framework
- Integration points
- Scalability considerations
- Disaster recovery implications

NLP Engine Selection

- Build vs. buy considerations
- Vendor comparison
- IBM-specific options
- Open-source alternatives
- Custom development needs

Data Preparation

- System metadata organization
- Application documentation digitization
- Business process mapping
- Training data collection
- Knowledge base development

Training and Fine-tuning

- IBM i domain adaptation
- Company-specific terminology
- Industry-specific requirements
- Technical dialect understanding
- Continuous learning framework

Security Framework

- Authentication mechanisms
- Authorization controls
- Audit logging requirements
- Data protection considerations
- Regulatory compliance implications

Integration Strategy

- IBM i Navigator integration
- Development environment connections
- Business application touchpoints
- Mobile and web interfaces
- API ecosystems

Change Management

- Skills transition planning
- Role evolution for IBM i professionals
- Training programs
- Communications strategy
- Organizational impact assessment

Phased Implementation Approach

- Phase 1: Read-only operations
- Phase 2: Minor system modifications
- Phase 3: Database operations
- Phase 4: Application development
- Phase 5: Full zero-code operations

Governance Model

- Oversight committee structure
- Quality assurance processes
- Approval workflows
- Exception handling
- Performance monitoring

Risk Management

- System integrity protections
- Rollback mechanisms
- Testing frameworks
- Performance degradation mitigations
- Business continuity planning

Cost-Benefit Analysis

- Implementation costs
- Training investments
- Infrastructure upgrades
- Productivity gains
- Error reduction savings
- Time-to-market improvements

Success Metrics Framework

- Development velocity KPIs
- System performance indicators
- User adoption measurements
- Business outcome impacts
- ROI calculation methodology

Challenges and considerations

Technical Challenges

- Integration complexity with legacy systems
- Performance overhead
- Reliability concerns
- Scaling for enterprise workloads
- Version control and change management

Quality Assurance Transformation

- Testing natural language interactions
- Validation methodologies
- Edge case identification
- Regression testing approaches
- Quality metrics evolution

Security Implications

- Attack surface changes
- Prompt injection risks
- Authentication challenges
- Audit trail requirements
- Data protection considerations

Ethical Considerations

- Bias in language models
- Transparency requirements
- Explainability challenges
- Accountability frameworks
- Human oversight necessity

Workforce Transformation

- Evolving roles for programmers
- New skills requirements
- Career path modifications
- Training and certification needs
- Organizational structure impacts

Regulatory Compliance

- Audit requirements
- Documentation changes
- Validation processes
- Industry-specific regulations
- International considerations

Data Privacy Concerns

- Information handling in NLP systems
- Confidentiality safeguards
- Data residency implications
- Anonymization techniques
- Consumer protection requirements

Business Process Adaptation

- Process documentation improvements
- Workflow modifications
- Decision authority frameworks
- Exception handling procedures
- Change management processes

Cost Considerations

- Implementation investments
- Ongoing operational costs
- Training expenditures
- Support requirements
- ROI timeline expectations

Limitations of Current Technology

- Domain-specific knowledge gaps
- Complex reasoning challenges
- Creative problem-solving boundaries
- Contextual understanding limitations
- Ethical judgment constraints

Human-in-the-Loop Requirements

- Review processes
- Override mechanisms
- Knowledge transfer approaches
- Continuous improvement feedback
- Accountability frameworks

Cultural Adaptation

- Organizational readiness assessment
- Leadership buy-in strategies
- User acceptance approaches
- Change resistance management
- Communication frameworks

Future Outlook

Evolution of NLP Technology

- Multimodal AI integration
- Reasoning capabilities enhancement
- Domain-specific specialization
- Quantum computing impacts
- Human-AI collaboration advancements

IBM i Platform Evolution

- IBM's strategic direction
- Database enhancements
- Cloud integration roadmap
- Hardware innovation impacts
- Ecosystem development

Converging Technology Trends

- Edge computing implications
- IoT integration opportunities
- Blockchain synergies
- Advanced analytics fusion
- Augmented reality interfaces

Skills of the Future

- Business-technology translation
- AI oversight competencies
- Prompt engineering expertise
- System design capabilities
- Ethical technology governance

Organizational Structure Evolution

- IT department transformation
- Citizen developer enablement
- Center of Excellence models
- Governance framework adaptation
- Cross-functional collaboration approaches

Beyond Zero-Code: Thought-Driven Computing

- Brain-computer interface possibilities
- Ambient computing concepts
- Anticipatory system behavior
- Emotion-aware interactions
- Autonomous optimization

Research Opportunities

- IBM i-specific NLP training methodologies
- Domain adaptation techniques
- Security framework innovations
- Performance optimization approaches
- User experience enhancements

Conclusion

Call to Action

- Community involvement opportunities
- Early adopter programs
- Research partnerships
- Feedback mechanisms
- Implementation support resources

Closing toughts

- Key takeaways

CD INVEST
BI · AI · IBMi

Thank You

For more information contact:

Koen Decorte
kdecorte@cdinvest.be