# DataLog

Datenbank-Programmiersprache
für deduktive Datenbanken

Proseminar
Programmierparadigmen und Sprachen

Nico Braunisch

# Inhalt

- DataLog
  - Motivation
  - Definition
  - Entstehung
  - Syntyx & Semantik
  - Bedeutung
  -
- CodeQuest

# Motivation

- Datenbank- und Logiksprachen sind oft langsam und unzureichend für Datenbankabfrage
- Kombination einer logischen Sprache mit einer deduktiven Datenbank
- Ziel: Erreichen möglichst kurzer Abfragedauer bei möglichst wenig Einschränkung an Sprachkonstruckte und Funktionsumfang

# Definition

- Answer Set Programming
  - Art der deklarativen Programmierung zur Lösung von Suchverfahren
  - Reduzierung auf ein einfaches Model
  - Erstellen einer Lösungsstrategie
- Logische Programmierung
  - Erstellen einer Menge von Axiomen
  - Auswertung über Abfragen der Axiomen

# Definition

- Deduktive Datenbank
  - Erweiterte relationale Datenbanken
  - Hinzufügen von Regeln und Fakten
  - Erlaubt Schlussfolgerung

# Entstehung

- Ursprung in der Entstehung des logischen Paradigma
- Wärend eines Seminars über Logik und Datenbanken
- Im Jahr 1978

# Entwickler

- Entwickelt von Herve Gallaire und Jack Minker
- Name wurde geprägt von David Maier

# Syntax & Semantik

- Ähnlich zu Prolog
- Beschränkt Konstrukte, die notwendig sind für die Datenbank-Abfragen
- Auswertung durch Fixpunktbrechnung
- Rekursionstiefe muss nicht explizit angeben werden

# Syntax

- Variable
  - <Großbuchstabe>{<Buchstabe> |<Zahlen>}
  - ”_” Annonyme Variable
- Konstante
  - <Kleinbuchstabe>{<Buchstabe> |<Zahlen>}
  - Oder eine Folge von Zahlen <Ziffer>{<Ziffer>}
- Term
  - <Variable>| <Konstante>

# Syntax

- Predikat Symbol
  - Kleinbuchstaben
- Atom
  - <Predikat Symbol> ”(”{<Terme>}”)”
  - Folge von (”nicht-zusammengesetzten”) Termen
  - $p(t_1 , . . . , t_n )$
  - n ist die Stelligkeit

# Syntax

- Klausel / Regel
  - $a \leftarrow b_1 \wedge \cdots \wedge b_m$
  - <Atom> [":-" {<Atom>}] "."
  - m ist die Stelligkeit
  - $\leftarrow$ = :- und $\wedge$ = ,
  - Klauseln ohne Atome rechts nennt man Fakt
  - Jede Variable im Kopf der Klausel muss auch im Rumpf vorkommen

# Syntax

- Wissensbasis / Programm
  - Menge von Klauseln
  - Ein Fakt für jedes Tupel der Datenbank
- Kommentare
  - % Ganze Zeile
  - /* Kommentar Anfang
  - */ Kommentar Ende

# Unterschied zu Prolog

- Argumente von Prädikaten durfen keine zusammengesetzte Term sein
- Negation und Rekursion müssen stratifiziert sein
- Auswertung erfolgt bottom-up
- Reihenfolge der Regeln spielt keine Rolle

# Semantik

- Wie in der Predikatenlogik
- Zustand der Datenbank wird als Interpretation aufgefasst
- Domäne und Konstanten sind dabei durch das System vorgegeben
- Herbrand-Interpretationen

# Bedeutung

- Zunächst nur theorethische Bedeutung
- Konnte sich als reine Datenbankabfragesprache nicht durchsetzen
- Keine Kommerzielle Lösung
- Dennoch praktisch Eingesetzt
- Weit verbreitet in Universitären Projekten

# Beispielsysteme

- ConceptBase konzeptionellen / Metamodellierung
- IRIS Datalog Erweiterung in Java
- DES genutz zur Schulung von DataLog
- DLV unterstütz disjuntive Normalform
- SecPAL Sicherheitsrichtlinen Dialeckt (MS Research)
- .QL objectorientierter Dialeckt
- Datalog leichtgewichtige DB-Sytem in LUA
- XSB: Logik & deductiv DB-System für Unix und Windows
- Bddbddb Abfrage & Analyse von Java Bytecode

# CodeQuest

- Entwickelt von Elnar Hajiyev, Mathieu Verbaere und Oege de Moor.
- 2006  University of Oxford,
- Inspiriert von JQuery (Eclipseplugin)

# CodeQuest

- Analyse von Objecktorientiernten Quellcodes
- Untersucht Beziehungen und Abhänigkeiten
- Ziel: entwicklung von generischer, fehlerfreier optimierter Software

# CodeQuest

- Mischung aus relationaler Datenbank und Datalog
- Speichert SyntaxBaum und Flusskontrolle in Datenbasis
- Balance zwischen Ausdruckskraft, Effizienz und Skalierbarkeit

# Quellcodeanalyse

- Überprüfung der Code Conventions
- Fehlersuche
- Refactoring
- Optimerung
- Komplexitätsanalyse

# Typ-Hierarchie

- Element
  - Paket
  - CU
  - Feld
  - Typ
    - Primitive
    - RefType
      - Schnitstelle
      - Klasse
- Block
  - Abrufbarer
    - Methode
    - Konstruktor
- Modifikator
- Parameter

# CodeQuest Queries

- type(T) T ist ein Type
- child(T, F ) F ist ein Kind von F
- field(F ) F ist ein Feld
- modifier(M, name) M ist ein Modifier mit name
- method(M ) M ist eine Methode
- writes(M, F ) M schreibt auf F
- subtypestar(T, T1 ) T indireckter Untetype T1
- ⋮

# Query Beispiel

- Beispiel 1

  query1(T,F) :- type(T), hasChild(T,F),field(F), hasStrModifier(F,'public'), NOT(hasStrModifier(F,'final')).

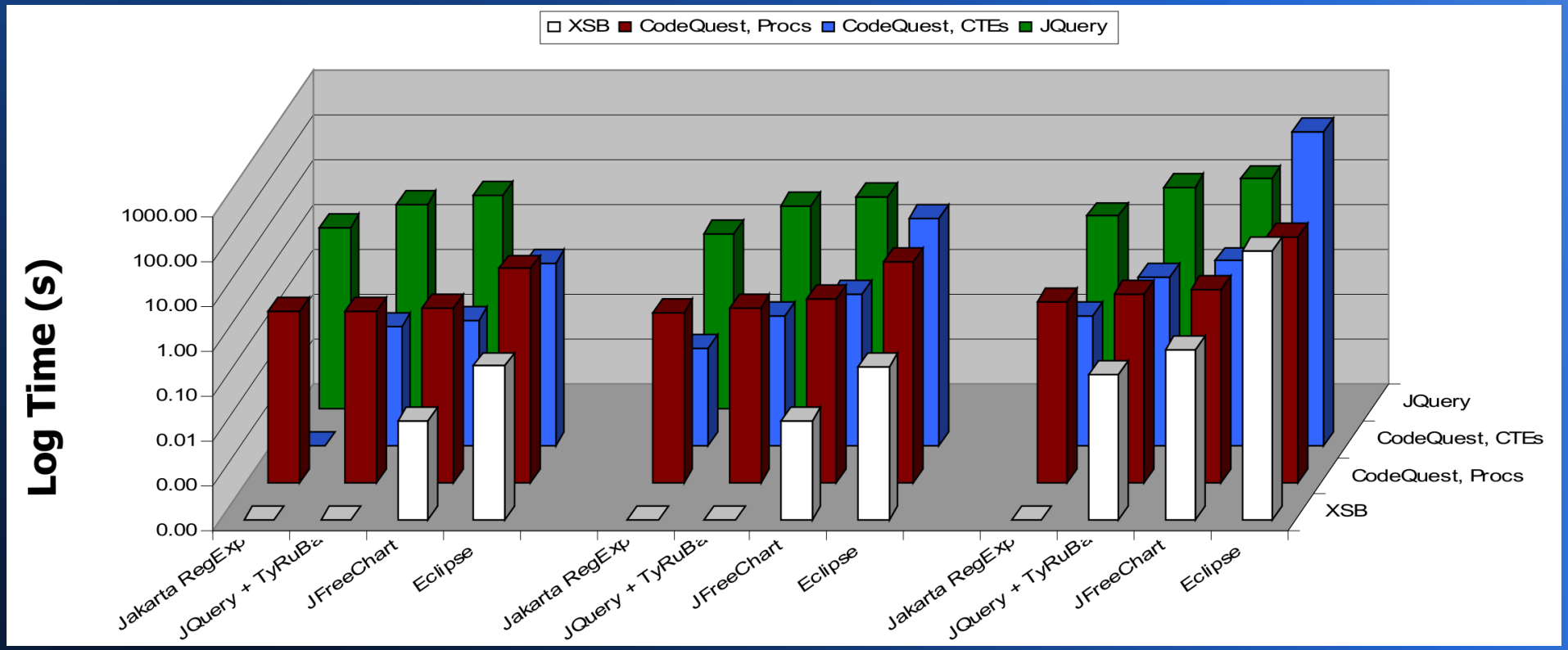
- Beispiel 2

  query2(M,T) :- method(M),writes(M,F), hasType(F,FT),hasSubtypeStar(T,FT).

- Beispiel 3

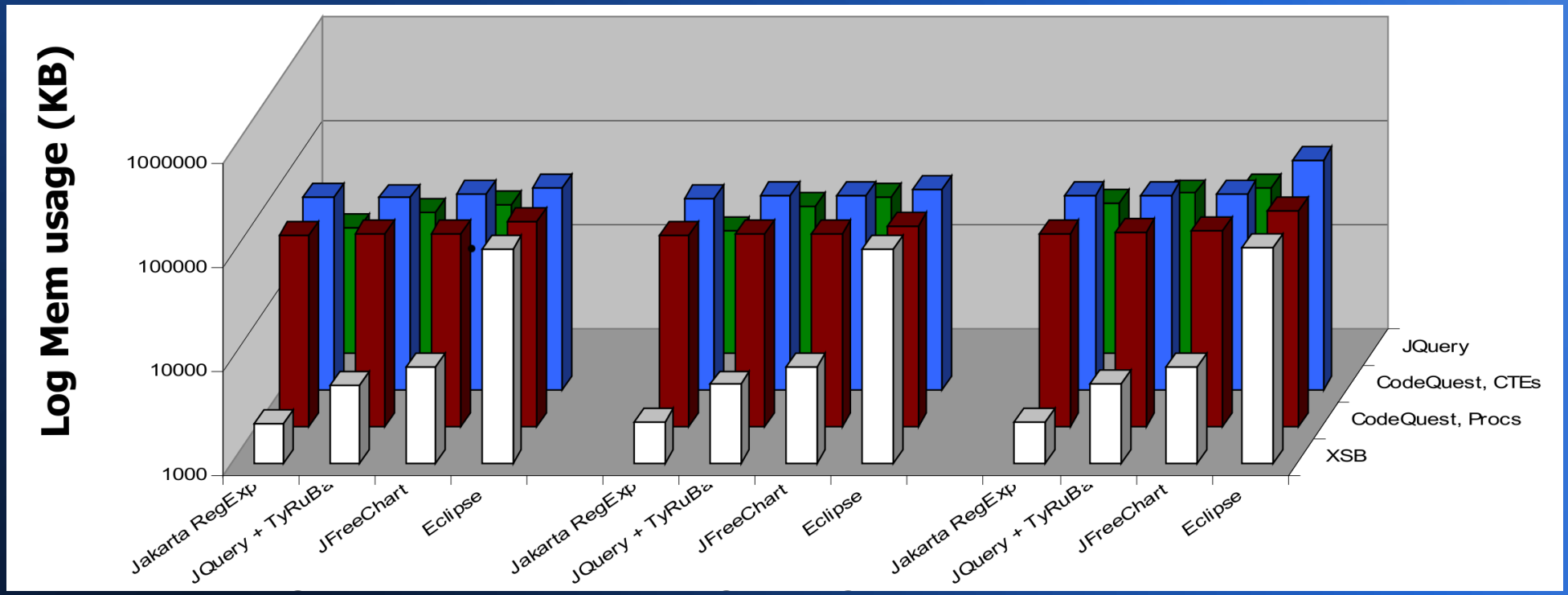
  query3(M1,M2):-overrides(M2,M1) ,hasStrModifier(M1,'abstract'), NOT(hasStrModifier(M2,'abstract')).

# Query Beispiel




Quelle: FinalCodeQuestPoster6

# Query Beispiel




Quelle: FinalCodeQuestPoster6

# Quellen

- www.progtools.comlab.ox.ac.uk/
  - Scalable Source Code Queries with Datalog
  - CodeQuest: Querying Source Code with DataLog
  - FinalCodeQuestPoster6
- www.wikipedia.org
  - Datalog
  - .QL
  - Stratifikation
  - Prolog
  - Answer set programming
  - Deduktion
- Logic: Datalog Syntax and Semantics
- Deduktive Datenbanken: 2. Datalog Syntax and Semantics

# Vielen Dank


Quelle: http://www.dresden-wissenschaft.de