EXAMINATION STATIONS

• Exam Friday
  • 3:30 – 4:50

• No note sheets/calculators/computers

• Practice solutions out after section tomorrow

• Good luck!
EXAM LENGTH

• Production v. Verification
  • Practice exam
• Short answer
  • Simplest answer possible
• Problems not necessarily in order of difficulty
GENERAL TOPICS

• Databases
  • Motivations and definitions
• Relational Databases
  • SQL
  • Relational Algebra
  • Datalog
• Semi-structured Data
  • Motivations and definitions
DATABASES

• Motivations
  • Collections of related files

• Databases vs. DBMS

• What is stored?

• What is the DBMS’ responsibility?
DATABASES

• Motivations
  • Collections of related files
• Databases vs. DBMS
• What is stored?
• What is the DBMS’ responsibility?
  • Data storage and manipulation
  • Black box thought
  • Physical data independence
RELATIONAL DATABASES

• Motivations
  • Breaking away from singular flat files
  • Why/how do we break up data?

• Data model
  • Schemas and keys
  • Records and attributes
  • Attribute types/typing
RELATIONAL DATABASES

• Primary keys
  • What are the constraints?
  • When do we select keys?
  • Multiple keys

• Foreign keys
  • Constraints vs. Joining

• Keys across different data
SQL STRUCTURE

• Flat tables
  • First normal form
  • Crosswalks and joins
  • Breaking up data into multiple relations
SQL CODE

• Create statements
  • Key declarations
  • Type declarations
• Insert/Delete statements
• Update statements
• Drop table
SQL CODE

- Select
- From
- Where
- Group by
- Having
- Order by
• Distinct (and relation to group by)
• Inner vs. Outer Joining
  • Left/Right/Full
• Nested loop semantics
  • Cross join with selection
• Self joins
  • Produce companies that produce gadgets and cameras
• **Aggregation**
  - `Count`, `sum`, `min`, `max`, `avg`
• **Null values**
  - `IS NOT null`
  - `Count(null)`
• **Where vs. Having**
SQL CODE

• Constructing Queries
  • FWGHOS

• Subqueries
  • In Select (Single attribute projection)
  • In From (subquery AS, WITH AS)
  • In Where (EXISTS, IN, ANY)
  • Correlated vs. Non-correlated
  • Un-nesting
  • Finding the Witness
SQL CODE

• Negation in subqueries

• Monotonicity
  • Definitions
  • Example
  • Difficulties and necessity of subqueries
RELATIONAL ALGEBRA

• Set vs. Bag semantics
  • Why bag?
• Query plans and RA expressions
• Operations (on relations, some with conditions)
  • Union, difference
  • Selection
  • Projection
  • Joins
RELATIONAL ALGEBRA

• Operations (on relations, some with conditions)
  • Union, difference
  • Selection
  • Projection
  • Joins
  • Duplicate elimination
  • Grouping
  • Sorting
RELATIONAL ALGEBRA

- Operations (on relations, some with conditions)
  - Union, difference
  - Selection
  - Projection
  - Joins (remember your conditions)
  - Duplicate elimination
  - Grouping
  - Sorting
RELATIONAL ALGEBRA

- How do we know SQL and RA are equally expressive?
  - Translating one to the other
  - Multiple RA expressions possible for same query
  - DBMS optimization
RELATIONAL ALGEBRA

• Producing RA expressions/trees
  • From queries
  • Visa-versa
• Bag vs. Set RA
  • Datalog is set semantic
DATALOG

- Queries which cannot be defined in RA
  - Recursive queries
- Expressing RA expressions in datalog
  - Set semantics (procedural)
    - “Simple, concise, elegant”
- Fixed point semantics
  - Recursion builds from base case
- Left/right/non-linear
DATALOG

• Logical framework
• Explicitly defined intermediate results

Terminology

• Facts and Rules
• Extensional vs. Intensional Predicates
• Head and body
• Head vs. Existential Variables
• Unsafe rules
DATALOG

• Writing Rules
  • Safety
  • Base cases
  • Aggregation and negation
  • Variable scope
  • Simple recursive queries
  • Converting from RA
SEMISTRUCTURED DATA

• Motivations
  • Transactional vs. Analytical Data
  • Data distribution
  • Consistency
  • Partition vs. Replication
  • Key-value storage -> Document Storage
JSON

• Gives structure to data
• Objects and collections
• Self described
• Separate and less constrained than SQL++
• Nested structure (non-first normal form)
ASTERIX DB

- Document-based
- NoSQL
- Semi-structured
- Over JSON objects
  - Constraints (types, no duplicates)
- SQL++
  - Description vs. Manipulation
ASTERIX DB

• Dataverse
  • Database – set of data currently working with

• Types
  • UUID – auto generated
  • Null vs. Missing
  • Nested collections
  • Open v. Closed
  • Required v. Optional fields
ASTERIX DB

- **Datasets**
  - Relations
  - Defined over a type
  - Must have a key
- **Indexes**
  - Over particular attributes
  - Speeds up 1-d selection (BTREE), 2-d selection (RTREE) and substring selection (KEYWORD)
ASTERIX DB

• SQL++
  • Heterogeneity
  • Unnesting
  • Nesting/Aggregation and non-first normal
  • Multi-value join
    • Supports one to many
  • Can often be represented in SQL
SEMISTRUCTURED

• Distributed systems
• Short-term analysis
• Lower set-up costs
• Higher query costs (often)
QUESTIONS

• That’s the material

• Things that will be on the exam
  • Short answer
  • SQL
    • Subquery
  • Datalog
  • Relational Algebra
QUESTIONS

- Probably on the exam
  - Design motivations
  - RA/SQL/Datalog equivalency
  - Data model construction
QUESTIONS

• Probably on the exam
  • Design motivations
  • RA/SQL/Datalog equivalency
  • Data model construction

• Not on the exam
  • Acronyms
  • SQL++ code
ADVICE

• Look through the exam first
  • Try and do easiest questions first
  • Short answer questions are worth equal amounts, varying difficulty

• Always be sure you understand the question
  • More “toy” examples on midterm
ADVICE

• Go through previous exams
  • Good judgement for questions
• Go through HW,OQ assignments
  • If I’ve asked you something before, I am certain that you should know how to do it
• Think about how null values/your assumptions impact the interpretation of the data