Tutorial 6: Exam preparation

Implementation of Databases (DBS2)

Arik Ermshaus

Tutorial Appointments

Week	Topic
16.10 - 20.10	-
23.10 - 27.10	Organisation, Exercise Sheet 1
30.10 - 03.11	Q&A
06.11 - 10.11	Q&A
13.11 - 17.11	Exercise Sheet 2
20.11 - 24.11	Q&A
27.11 - 01.12	Q&A
04.12 - 08.12	Exercise Sheet 3
11.12 - 15.12	Q&A
18.12 - 22.12	Q&A
25.12 - 29.12	-
01.01 - 05.01	-
08.01 - 12.01	Exercise Sheet 4
15.01 - 19.01	Q&A
22.01 - 26.01	Q&A
29.01 - 02.02	Exercise Sheet 5
05.02 - 09.02	Q&A
12.02 - 16.02	**Exam preparation**

Table of Contents

Task 1: Query Optimization

(i) Write following query as algebraic term:

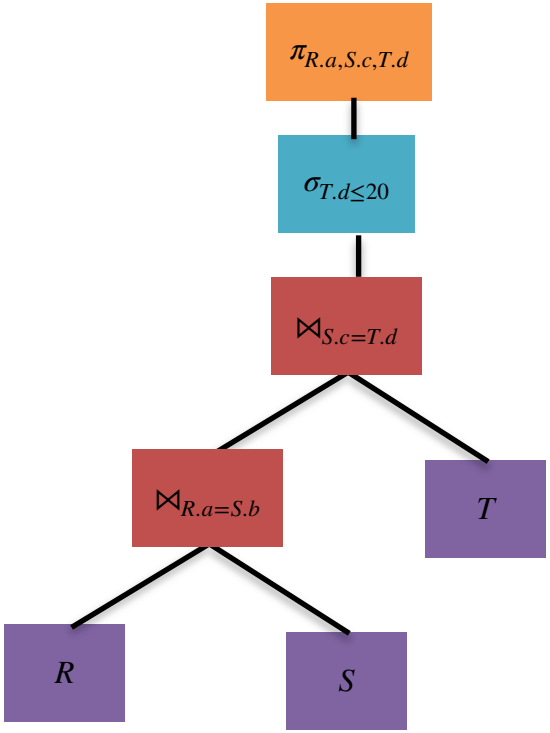
SELECT R.a, S.c **FROM** R, S **WHERE** R.a = S.b **AND** S.c > 10

- $\pi_{R.a,S.c}(\sigma_{R.a=S.b \wedge S.c>10}(R \times S))$

Task 1: Query Optimization

(ii) Draw possible operator tree for query:

SELECT R.a, S.c, T.d **FROM** R, S, T **WHERE** R.a = S.b **AND** S.c = T.d **AND** T.d <= 20



Task 1: Query Optimization

(iii) Minimize amount of operators in algebraic term:

$$\pi_{T.d,U.f}(\sigma_{T.d!=5}(\pi_{T.d,U.e,U.f}(\sigma_{T.d=U.e}(\pi_{T.d,U.e,U.f}(T \times U)))))$$

- $\pi_{T.d,U.f}(\sigma_{T.d=U.e \wedge T.d \neq 5}(T \times U))$

Task 1: Query Optimization

(iv) Does minimised algebraic term lead to minimal intermediate results? Provide explanation.

- No, not necessarily
- Focus is on reducing complexity of expression, not on size of data being processed
- Optimizations like breaking conjunctions, projections can produce more operators but smaller intermediate results
- Actual execution cost depends on physical factors like size of the data sets, available indices

Task 1: Query Optimization

(v) Optimize algebraic term according to “rule-based optimizer”:

$$\pi_{T.d,U.f}(\sigma_{T.d!=5}(\pi_{T.d,U.e,U.f}(\sigma_{T.d=U.e}(\pi_{T.d,U.e,U.f}(T \times U)))))$$

- Break / push down conditions:

  $$\pi_{T.d,U.f}(\sigma_{T.d=U.e}(\sigma_{T.d!=5}(\pi_{T.d}(\pi_{T.d}(T))) \times \pi_{U.e,U.f}(\pi_{U.e,U.f}(U))))$$

- Merge operators:

  $$\pi_{T.d,U.f}(\sigma_{T.d!=5}(\pi_{T.d}(T)) \bowtie_{T.d=U.e} \pi_{U.e,U.f}(U))$$

Task 2: Cost Estimation

Consider relations R, S, T and statistics. Assume uniform distribution.

R(a,b)	S(b,c)	T(c,d)
|R| = 300	|S| = 500	|T| = 900
v(R, a) = 30		
v(R, b) = 60	v(S, b) = 50	
	v(S, c) = 100	v(T, c) = 90
		v(T, d) = 60

(i) Unbiasedly estimate cardinality of algebraic term:

$$|R \bowtie_{R.b=S.b} S \bowtie_{S.c=T.c} T| = \frac{|R| \cdot |S| \cdot |T|}{max(v(R,b), v(S,b)) \cdot max(v(S,c), v(T,c))} = \frac{300 \cdot 500 \cdot 900}{60 \cdot 100} = 22500$$

Task 2: Cost Estimation

Consider relations R, S, T and statistics. Assume uniform distribution.

R(a,b)	S(b,c)	T(c,d)
|R| = 300	|S| = 500	|T| = 900
v(R, a) = 30		
v(R, b) = 60	v(S, b) = 50	
	v(S, c) = 100	v(T, c) = 90
		v(T, d) = 60

(ii) Unbiasedly estimate cardinality of algebraic term:

$$|\sigma_{a=10}(R)| = \frac{|R|}{v(R,a)} = \frac{300}{30} = 10$$

Task 2: Cost Estimation

Consider relations R, S, T and statistics. Assume uniform distribution.

R(a,b)	S(b,c)	T(c,d)
|R| = 300	|S| = 500	|T| = 900
v(R, a) = 30		
v(R, b) = 60	v(S, b) = 50	
	v(S, c) = 100	v(T, c) = 90
		v(T, d) = 60

(iii) Unbiasedly estimate cardinality of algebraic term:

$$|\sigma_{d>10}(T)| = \frac{|T|}{const} = \frac{900}{3} = 300$$ (with const = 3)

Task 2: Cost Estimation

Consider relations R, S, T and statistics. Assume uniform distribution.

R(a,b)	S(b,c)	T(c,d)
|R| = 300	|S| = 500	|T| = 900
v(R, a) = 30		
v(R, b) = 60	v(S, b) = 50	
	v(S, c) = 100	v(T, c) = 90
		v(T, d) = 60

(iv) Unbiasedly estimate cardinality of algebraic term:

$$|S \times T| = |S| \cdot |T| = 450000$$

Task 2: Cost Estimation

Consider relations R, S, T and statistics. Assume uniform distribution.

R(a,b)	S(b,c)	T(c,d)
|R| = 300	|S| = 500	|T| = 900
v(R, a) = 30		
v(R, b) = 60	v(S, b) = 50	
	v(S, c) = 100	v(T, c) = 90
		v(T, d) = 60

(v) Unbiasedly estimate cardinality of algebraic term:

$$|\sigma_{c>1 \wedge d=2}(T)| = \frac{|T|}{3 \cdot v(T,d)} = \frac{900}{3 \cdot 60} = 5$$

Task 3: Joins

Implement “Join” query operator as blocked execution.

```cpp
bool Join::open()
{
    source2->open();

    // store blocks of joined records
    std::vector<std::string> block_ids;

    // create first block
    std::shared_ptr<Block> block = buffer_manager->fix_block(buffer_manager->create_new_block());
    block_ids.push_back(block->get_block_id());

    // read first record (of 2nd source)
    std::shared_ptr<Record> record2 = source2->next();
    int current_record = 0;

    while (record2)
    {
        // get join attribute 2
        Record::Attribute attribute2;

        if (attribute_types2[attribute_position2] == "int")
            attribute2 = record2->get_integer_attribute(attribute_position2);
        else if (attribute_types2[attribute_position2] == "string")
            attribute2 = record2->get_string_attribute(attribute_position2);
        else if (attribute_types2[attribute_position2] == "bool")
            attribute2 = record2->get_boolean_attribute(attribute_position2);

        // create selection operator
        std::shared_ptr<Selection> selection = std::make_shared<Selection>(Selection(
                buffer_manager, source1, attribute_position1,
                attribute_types1[attribute_position1],attribute2, comparator)
        );

        // read matching records
        selection->open();
        std::shared_ptr<Record> record1 = selection->next();

        // ...
```

```cpp
    // ...
    while (record1)
    {
        // join records
        std::vector<Record::Attribute> attributes;

        for (int i = 0; i < attribute_types1.size(); i++)
        {
            if (attribute_types1[i] == "int")
                attributes.push_back(record1->get_integer_attribute(i));
            else if (attribute_types1[i] == "string")
                attributes.push_back(record1->get_string_attribute(i));
            else if (attribute_types1[i] == "bool")
                attributes.push_back(record1->get_boolean_attribute(i));
        }

        for (int i = 0; i < attribute_types2.size(); i++)
        {
            if (attribute_types2[i] == "int")
                attributes.push_back(record2->get_integer_attribute(i));
            else if (attribute_types2[i] == "string")
                attributes.push_back(record2->get_string_attribute(i));
            else if (attribute_types2[i] == "bool")
                attributes.push_back(record2->get_boolean_attribute(i));
        }

        std::shared_ptr<Record> record = block->add_record(attributes);

        // edge case: block is full
        if (!record) {
            // create next block
            buffer_manager->unfix_block(block->get_block_id());
            block = buffer_manager->fix_block(buffer_manager->create_new_block());
            block_ids.push_back(block->get_block_id());

            assert(block->add_record(attributes));
            current_record = 1;
        }

        // go to next record position
        if (current_record == Block::MAX_RECORDS - 1)
        {
            current_record = 0;
            // create next block
            buffer_manager->unfix_block(block->get_block_id());
            block = buffer_manager->fix_block(buffer_manager->create_new_block());
            block_ids.push_back(block->get_block_id());

        } else
            ++current_record;

        record1 = selection->next();
    }

    selection->close();
    record2 = source2->next();
}

buffer_manager->unfix_block(block->get_block_id());

// create and open intermediate table
results = std::make_shared<Table>(Table(buffer_manager, block_ids));
return results->open();
}
```

```cpp
std::shared_ptr<Record> Join::next()
{
    return results->next();
}

bool Join::close()
{
    source2->close();

    // delete intermediate table blocks
    for (auto const& block_id : block_ids)
        buffer_manager->erase_block(block_id);

    return results->close();
}
```

Table of Contents

- Solutions of Exercise Sheet 5
- **Exam preparation**

Exam Details

- First exam
  - Date: Monday, 26. February 2024 12:00 o'clock
  - Room: 0'307 and 0'311 (RUD26)
  - Please arrive between 11:30 and 12:00 o'clock!

- Second exam
  - Date: Monday, 8. April 2024 12:00 o'clock
  - Room: 0'110 (RUD26)
  - Please arrive between 11:30 and 12:00 o'clock!

- Checklist for you!
  - Registration in Agnes
  - Carry national ID card and student ID card
  - Blue or black pen (no other aids allowed)

Topics: An overview

- Disks, records and blocks
- Caching
- File structures
- Hashing
- B trees
- Multi-dimensional index structures
- Query execution
- Joins
- Query optimization
- Recovery
- Synchronization

Exam preparation: Basics

- Learn with lecture and UE slides provided on websites

- Definitions
  - Can be be asked for
  - Be prepared to use and explain them

- User / DB / real-word scenarios
  - May be given with custom assumptions
  - Be able to calculate and explain different aspects

- Concepts
  - Be prepared to explain (properties, advantages / disadvantages, edge cases)
  - Be able to link them to other concepts

- Algorithms
  - Be prepared to implement, describe and analyse them
  - Be able to perform them step-by-step for a provided example

Disclaimer: These tips are for orientation only and not necessarily complete

Example Questions: Disks, Records and Blocks

- What is a Cylinder?
- Calculate the latency for …
- What advantage does RAID5 have over RAID0?
- Calculate the record size for …
- Describe how a record is inserted into a block (with directory)?
- …

Disclaimer: These tips are for orientation only and not necessarily complete

Example Questions: Buffer Manager

- Explain the concept of fixing and unfixing blocks
- Which of the strategies … is most suited for a BNL join?
- Perform cache replacement using CLOCK and …
- How can cache consistency be provided in multi-TX systems?
- What is semantic caching?
- …

Disclaimer: These tips are for orientation only and not necessarily complete

Example Questions: B trees

- Define the conditions for an internal B tree node
- Insert … into the following B+ tree
- Which blocks are loaded searching for … in the following B+ tree
- What is the difference between a B* and B+ tree?
- Given scenario … Should you use a B+ tree or hashing with …. ?
- …

Disclaimer: These tips are for orientation only and not necessarily complete

Example Questions: MDIS and Query Execution

- Given scenario … Should you use two independent indices or one composite index with …. ?
- Insert the attributes and tuple ID … into the grid file …
- Split the R-tree node … using a linear split
- Implement a GROUP BY using C++ code, given …
- Rewrite the blocked filter from … into a pipelined one, using …
- …

Disclaimer: These tips are for orientation only and not necessarily complete

Example Questions: Query Optimization

- Define cascading selections
- Optimize SQL query … using “rule-based” optimizer
- Choose join order for … using dynamic programming
- Create equi-depth histogram for table …
- Estimate cost for SQL query … based on histograms …
- …

Disclaimer: These tips are for orientation only and not necessarily complete

Example Questions: Recovery and Synchronisation

- Define a transaction
- Use the undo log to restore the DB state before the crash …
- Is the following schedule … serial?
- Test if schedule … is conflict-serializable
- What is difference between pessimistic and optimistic scheduling?
- …

Disclaimer: These tips are for orientation only and not necessarily complete