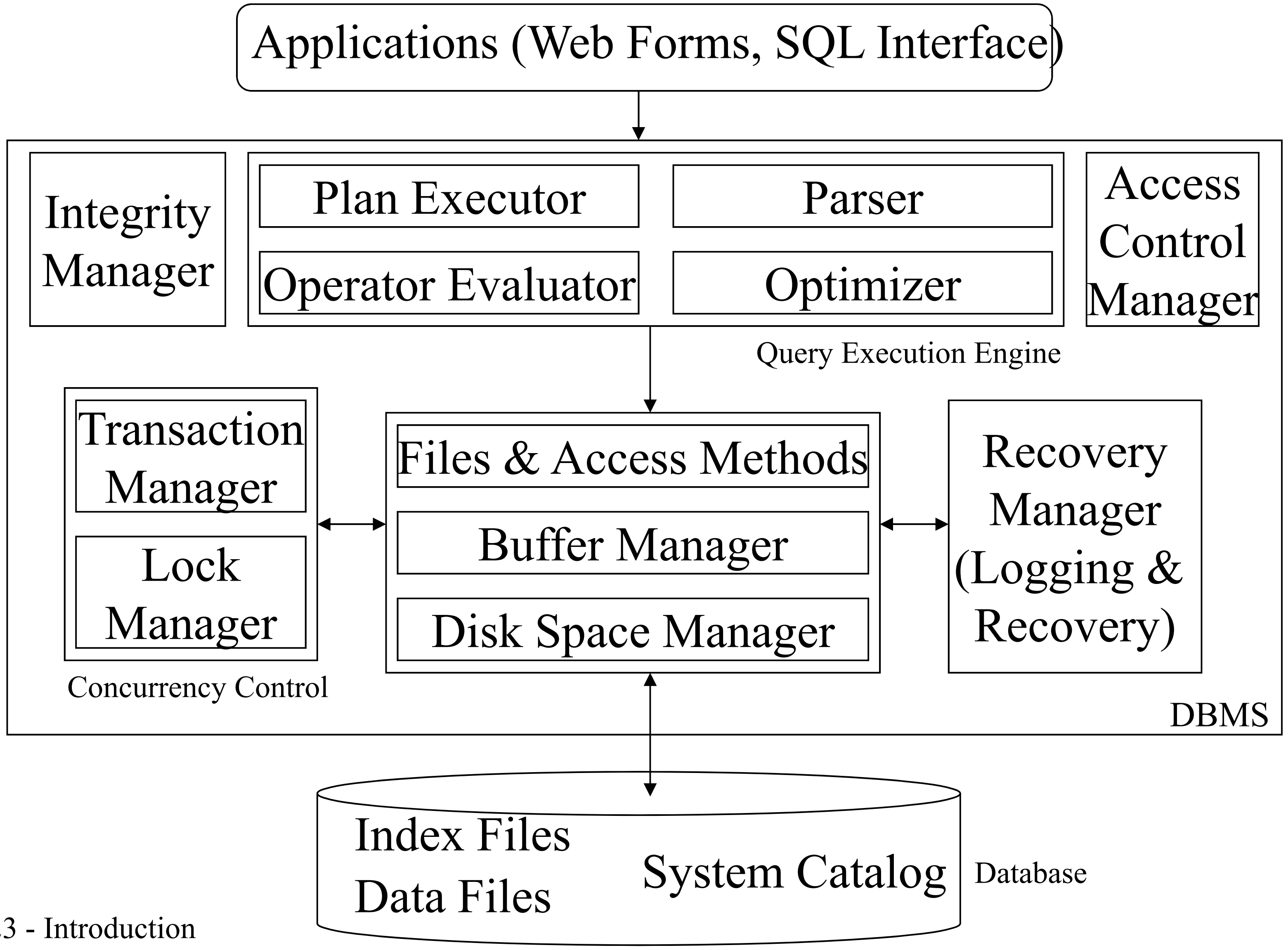
Review

Architecture of a DBMS



What good is this course?

Query optimization

- Consider the relations R(A, B, C), S(C, D, E) and T(E, F, G), with primary keys A, C and E respectively. Assume that R has 10000 tuples, S has 10000 tuples and T has 10000 tuples. For simplicity, assume that all attributes are 4 bytes, and that each page is 100 bytes. Consider the queries:

```
QA: SELECT A, B
       FROM R, S
       WHERE R.C = S.C;
```

```
QB: SELECT  C, COUNT(B)
       FROM R, S
       WHERE R.C = S.C
       GROUP BY  C;
```

```
QC: SELECT *
       FROM R, S, T
       WHERE R.C = S.C
       AND S.E=T.E;
```

- What is the output size (in terms of number of tuples and pages) for query QA?
- What is the output size (in terms of number of tuples) of QB?

- Consider query QB. Suppose the buffer size is 100 pages. Compute the cost (in terms of I/O) of processing QB under nested-loops join and sort-merge join algorithms. For simplicity, assume all intermediate results are written to disk; moreover, you should keep any intermediate results as small as possible, i.e., you should remove any unnecessary attributes as soon as possible. Ignore the cost to write out the final output.

- For query QC, a "new" query optimizer generates the following two plans:
  - Plan 1: (R JOIN S) JOIN T
  - Plan 2: I1 JOIN I2 where I1 = R JOIN S and I2 = S JOIN T
- What is the join condition between I1 and I2? Are there situations where Plan 2 is superior over plan 1? Justify your answer.

Concurrency Control

- Serializability concept is for committed transactions only

Concurrency Control

- Suppose we run the following 5 transactions using the validation-based protocol. There are no other transactions in the system. Suppose the database has only the set of given 7 objects: a, b, c, d, e, f, g. For transaction T_5, the values XXX and YYY represent unknown values. The following table lists the transactions involved, together with their read and write sets:

Transaction	Read Set	Write Set
T_1	{a, b}	{c}
T_2	{a, b}	{c, d}
T_3	{d}	{e}
T_4	{f}	{e, g}
T_5	{XXX}	{YYY}

- Consider the following schedule of events. For transaction with id i, S_i stands for start, V_i for validation and F_i for finish.

$$H = S_1; S_2; V_1; F_1; S_3; V_2; S_4; S_5; V_3; V_4; F_2; F_3; V_5; F_4; F_5$$

Concurrency Control

- Ignore T5 for now, which transactions validate successfully?

Concurrency control

- Suppose transaction T_5 validates successfully. What are the largest possible sets of values of XXX and YYY for this to happen?

- Suppose transaction T_5 does not validate successfully. What is the smallest possible write set of value YYY for this to happen?

Recovery

- A DBMS uses two phase locking as the concurrency control mechanism, and there are only read and write locks. To manage recovery, it uses UNDO/REDO logging with non-quiescent checkpoints (with checkpoint starts and checkpoint ends). In the log, the format of each record is <tid, objectID, old value, new value>. The log contents are shown below (sequence is ordered left to right, top to bottom, so < T1, start> is the first entry in the sequence and < T3, Z, 40, 80> is the last).

  < T1, start>; < T1, X, 5, 10>; < T2 start>; < T2 X, 10, 20>;< T1 commit>;

  < T2, Y, 30, 60>; < chkpt start; L>; < T2, W, 35, 70>; < T3, start>; < T3, W, 70, 40>;

  < chkpt end>; < T2, Z, 20, 40>; < T2, commit>; < T3, Z, 40, 80>

  where L is the list of active participants.

Recovery

- Is it possible for the log to have the above contents, given our assumptions about the system? Explain briefly. If the answer is no, which is the first "impossible" log entry? Why? Remove that entry. Is it possible for the log to contain the new sequence? Again explain why or why not. Repeat until you get a possible sequence of log entries. (You only need to indicate which log(s) is(are) to be removed, and why. There is no need to copy all the log records again.)

Recovery

- For the sequence of entries you got in (1), what are the possible values of X,Y,W,Z after the last of these log records is written to disk and before recovery? Fill your values in the table below (If you think X can possibly be 5, 10, or 20, then write all these numbers).

Recovery

- When the recovery manager runs after a crash, it can execute the following types of actions:
  - write(Z, v): This action stores the value v into database object Z.
  - abort(Tj): This action writes an abort Tj record into the log. (The record is not written to the log until all Tj writes generated during recovery have been flushed to disk.)

- A tanklDB1 system uses UNDO logging only. Given the contents of the log below, list the actions performed by the recovery manager (using the two action types given above). In the log shown here, the format of each record is <Tid, ObjectID, OldValue>. The oldest record is shown in the left most position. The second log record written is the second one. The last record written before the crash is the last one.

  <T1, start>; <T1, X, 10>; <T2, start>; <T2, Z, 12>; <T2, Y, 20>; <T2, commit>

Recovery

- The tanklDB2 system uses REDO logging only. The contents of the log at recovery time are given below. List the actions performed by the recovery manager. The format of each log record is <Tid, ObjectID, NewValue>.

  <T1, start>; <T1, X, 10>; <T1, commit> ; <T2, start>; <T2, Z, 12>; <T2, Y, 20>; <T2, commit>

- The tanklDB3 system uses UNDO/REDO logging with non-quiesce checkpointing. Given the contents of the log below, list the actions performed by the recovery manager. The format of each log record is <Tid, ObjectID, OldValue, NewValue>.

  <T1, start>; <T1, X, 10, 11>; <T1, commit> ; <Start Checkpoint> ; <T2, start>; <T2, Y, 20, 21>; <T3. start> ; <T3, Z, 30, 31> ; <End Checkpoint> ; <T4, start> ; <T3, commit> ; <T4, W, 40, 41>, <T2, Z, 80, 30>

Final Assessment

During the period when Einstein was active as a professor, one of his students came to him and said: "The questions of this year's exam are the same as last years!" "True," Einstein said, "but this year all answers are different."

- Face-to-face
  - 27th April, 1-3pm, MSPH6 (please verify this yourself)
- Open book
- Topics:
  - Query Optimization, Concurrency Control and Recovery (L7-L12)
  - While you will not be tested on tree structures or hashing or sorting algorithm or join algorithms, you are expected to know the costing (since they are part of cost model in optimizer)
- Bring along an authorized calculator & blank papers
- You can also use a laptop/iPAD BUT NO network access

Final Assessment

- Total: 30 marks
- 2 sections (120 minutes)
  - Section A (13 marks)
    - 3 open questions
  - Section B (17 marks)
    - 17 questions
    - Similar style as our mid-term

Viewing of Papers

- Optional
- 4th May (one week after exam)
- 9am to 4pm (except 11.30-1pm)
- COM1 03-23
- At most 7 persons each time; at most 10 minutes

This is not the end.
It is not even the beginning of the end.
It is, perhaps, the end of the beginning.

- Winston Churchill