Write all of your answers directly on this paper. Be sure to clearly indicate your final answer for each question. Also, be sure to state any assumptions that you are making in your answers.
Question 1 [15 parts] True/False Statements

State if the following statements are TRUE or FALSE. Please write your answers in the boxes below. No explanation is needed.

(1) For a given SQL query, there may be several equivalent relational algebra expressions, and for each such expression, there can be multiple query plans of different costs.

(2) Consider relational operators $\sigma$ and $-$. For any union compatible relations $R_1$ and $R_2$ and any predicate $p$, $\sigma_p(R_1 - R_2) \equiv \sigma_p(R_1) - R_2$ (‘$\equiv$’ means that the left expression and the right expression always return the same answer).

(3) Consider the relation DIRECTFLIGHTS(from, to). We want to find the cheapest flight from Boston to San Francisco with any number of stops in between. We can obtain the answer using a First Order (FO) query.

(4) On average, repeated random IO’s are faster than sequential IO’s because random IO’s tend to access different cylinders and therefore cause less contention.

(5) It is a good idea to create as many indexes as possible to expedite query processing because there are no disadvantage of having many indexes.

(6) Consider relations $R(a, b)$ and $S(b, c)$. A hash join can be used to efficiently implement $R \bowtie S$ with the join condition $R.a > S.c$.

(7) A clustered B+Tree is always faster than a file scan to retrieve the records matching a predicate.

(8) A B+Tree on <age, salary, department> may answer a selection query, “age=20 AND department=‘CS’”, more efficiently than a file scan.

(9) Using an unclustered B+Tree index on age to retrieve records in sorted order of age is faster than performing a two-pass external merge-sort (assuming that we have enough memory to do so).
(10) Consider two relations $R(A, B)$ and $S(A, D)$. It is known that $R.A$ is a foreign key referencing $S.A$, and that $S.A$ is the primary key of relation $S$. Then the estimated size of $R \bowtie S$ is $|R|$.

(11) If a schedule of transactions is serializable, it must also be conflict serializable.

(12) The Strict Two-Phase Locking (Strict 2PL) protocol allows only schedules that are conflict serializable and recoverable.

(13) If we follow the two-phase locking proposal (2PL) strictly, we won’t run into deadlocks.

(14) The phantom problem may occur if we allow transaction $T_1$ to run the same query twice on relation $R$ and, at the same time, allow transaction $T_2$ to update a tuple in $R$.

(15) When a database system restarts and runs the recovery process, it runs the Analysis, Undo, and Redo, phases in the specified order.
Answers:
(1) True
(2) True. It is easy to prove that for each tuple t in \(\sigma_p(E1 \cdot E2)\), it also exists in \(\sigma_p(E1) - E2\), and vice versa.
(3) False. Path queries cannot be expressed in FO.
(4) False. Repeated random IOs are much slower than sequential IOs due to the repeated overheads of the seek and rotational delay.
(5) False. There are disadvantages of creating too many indexes. Every time when we insert a new tuple, we need to update all indexes. Having too many indexes slows down insertions.
(6) False. A hash join can’t handle inequality predicates.
(7) False. (it depends on the selectivity)
(8) True
(9) False. To use an unclustered index to retrieve tuples in sorted order, we have to follow the (RID) pointers of the data entries as we scan them in the leaf nodes. In the worst case, we have to pay 1 random I/O per data entry.
(10) True
(11) False. It is the opposite: if a schedule is conflict serializable, it is serializable.
(12) True
(13) False. Deadlocks occur even if we follow the two-phase locking proposal (2PL) strictly.
(14) True
(15) False: It should be Analysis, Redo, and Undo.