

Database Design and Implementation
Chapter 10 Conceptual Exercises

Questions	Answers
10.1.a.	<p>To calculate the number of disk accesses required for the given relational algebra query, we need to break down the query into its components and calculate the disk accesses for each step.</p> <ol style="list-style-type: none"> 1. $T1 = \text{select}(\text{DEPT}, \text{DName}='math')$ - This is a selection operation on the DEPT relation where $\text{DName}='math'$. Assuming an index on the DName attribute, the number of disk accesses for this operation would be the number of blocks needed to store all tuples with $\text{DName}='math'$. 2. $T2 = \text{select}(\text{STUDENT}, \text{GradYear}=2018)$ - Similarly, this is a selection operation on the STUDENT relation where $\text{GradYear}=2018$. Assuming an index on the GradYear attribute, the number of disk accesses for this operation would be the number of blocks needed to store all tuples with $\text{GradYear}=2018$. 3. $\text{product}(T1, T2)$ - This is a cartesian product between T1 and T2. The number of disk accesses for a cartesian product is the product of the number of blocks in T1 and T2. <p>So, to calculate the total number of disk accesses: Disk accesses for T1 + Disk accesses for T2 + Disk accesses for $\text{product}(T1, T2)$</p>
10.1.b.	<p>If the arguments to the product operation are exchanged, we would calculate the number of disk accesses as follows:</p> <p>Disk accesses for T2 + Disk accesses for T1 + Disk accesses for $\text{product}(T2, T1)$</p>
10.2.	<p>$B(s)$: The number of blocks that contain the tuples of the relation s. $R(s)$: The number of tuples in the relation s. $V(s, F)$: The number of distinct values that a field F takes in relation s.</p>
10.3.	<p>To demonstrate this, we typically use the formula for calculating the cost of a nested loop join. Notice that, this is equivalent to the relational algebra product.</p> $\text{Cost} = B(R) + B(R) * R(S)$ <p>However, the logic would involve showing that swapping the inner and outer relations changes the number of block accesses due to the different sizes of the relations.</p>
10.4.	<p>The total number of block accesses required for the Cartesian product with STUDENT as the outer scan would be 94,500.</p>
10.5.a.	<p>The plan for this query would involve scans and selections for STUDENT, COURSE, ENROLL, and SECTION, with product operations to join these tables on the specified conditions and a final project operation for SName and Grade.</p>
10.5.b.	<p>The plan for this query would involve a scan of STUDENT and ENROLL, selection for MajorId=10 and Grade='C', and a product operation to join them on SId = StudentId.</p>

Database Design and Implementation
Chapter 10 Conceptual Exercises

10.6.	<p>The planner must ensure that:</p> <ul style="list-style-type: none">• All the fields specified in the WHERE clause exist in the tables being joined.• The joins are on valid keys and foreign keys.• The tables in the FROM clause are available and accessible.• Data types are compatible in comparisons and joins.
10.7.a.	<p>For the insert statement, the planner must verify that:</p> <ul style="list-style-type: none">• SId, SName, GradYear, and MajorId are valid columns in STUDENT.• The values conform to the data types and constraints of the columns.• SId does not violate any uniqueness constraints.
10.7.b.	<p>For the delete statement, checks include:</p> <ul style="list-style-type: none">• Existence and access to STUDENT and ENROLL tables.• Validity of the subquery and its relation to the main query.• MajorId is a valid field, and the subquery correctly identifies records for deletion.
10.7.c.	<p>For the update statement:</p> <ul style="list-style-type: none">• Confirm that MajorId and GradYear are valid columns in STUDENT.• Check that the subquery involving DEPT is valid.• Ensure that updating GradYear maintains any constraints on the column.