Umeå University
Department of Computing Science
TDBC86 — Database Concepts
Examination: December 21, 2001

Name and ID number: ____

Signature: ____

1. All answers must be written in English.

2. A pocket engineering calculator (miniräknare) and English/X - X/English dictionary may be used. No other help materials are allowed.

3. The examination consists of two parts. In Part I, which includes Problems 1–5, on pages 1–7, the answers **must** be written on the sheets provided. In general, only the answers will be checked for these questions; derivations need not be provided and will not be evaluated. In Part II, which includes Problems 6–12, the answers must be written on separate sheets. If appropriate for Problems 6–12, work showing how the answers were obtained will be evaluated, and in some cases is required.

4. You must turn in this cover sheet, the answer sheets for Part I (pages 1–7), and the pages containing your answers for Problems 6–12. Collate the sheets in numerical order of the problems, and write on only one side of the paper.

5. The examination has a total of 800 points, which constitutes 80% of the course grade. The other 20% is based upon the obligatory exercises, as explained in the course syllabus.

5. For Problems 6–12, you have the choice, for each part, to do the problem, or to skip it for partial credit. In the table below, place an X in the position for any problem for which you have attempted a solution, and which you wish to have graded. It is extremely important that you fill in this table properly, because of the following option. For any box which is left blank, the associated question will not be graded, and you will instead be awarded 15% of the points for that question. Your decision to leave a box blank is definitive, so be very careful. For example, If you leave box 8(b) blank, your answer to that question will not be graded, even if it is completely correct. On the other hand, if you place an X in box 8(b), but provide no answer whatsoever to that question, you will not receive 15% of the points for that question. It is strongly recommended that you use a pencil, in case you change your mind! *Note that Problems 1–5 do not carry this option. No points are awarded for skipping them.*

Prob	6	7	8	9	10	11	12
(a)							
(b)							
(c)							
(d)							
(e)							

Name and ID number: ______________________________

(1: 80 points total) For each of the following statements, circle the **T** is the assertion is true, and the **F** if it is false. The point total will be computed as $4 \times C - 2 \times I$ with C = number of correct answers and I = number of incorrect answers; however, the score may never be less than zero. A blank answer is neither in C nor in I.

T F (a) In a relational schema constrained by functional dependencies, every candidate key is a superkey.

T F (b) In SQL schema declaration using **Create Table**, it is not possible to declare candidate keys which are not the primary key.

T F (c) If $\mathbf{K}$ is a candidate key for a relation schema $R[\mathbf{A}]$ and $\mathbf{K} \subseteq \mathbf{B} \subseteq \mathbf{A}$, then $\mathbf{K}$ is always a candidate key for $\pi_{\mathbf{B}}(R[\mathbf{A}]) = R[\mathbf{B}]$.

T F (d) A foreign key may never be the primary key in the relation in which it occurs.

T F (e) A foreign key may never contain null values.

T F (f) Every relational schema constrained by functional dependencies has a lossless decomposition into 3NF.

T F (g) Every relational schema constrained by functional dependencies has a lossless decomposition into BCNF.

T F (h) Hashing is an effective means for the realization of a dense secondary index.

T F (i) A file may have both a primary index and a clustering index.

T F (j) It is impossible to write unsafe queries in the relational algebra.

T F (k) One purpose of the system catalog is to maintain statistical information on the size of relations.

T F (l) With the pure deferred update strategy, transactions never need be rolled back.

T F (m) With the pure immediate update strategy, it is not necessary to write after images to the log file.

T F (n) In a no-steal approach, updates made to data items must be written to disk when the transaction commits.

T F (o) A steal/no-force caching strategy is likely to be more efficient in terms of system resource use than a no-steal/force strategy.

T F (p) An effective way to avoid livelock in the wound-wait strategy is to restart aborted transactions with their old timestamps.

T F (q) In the authorization model of SQL, user *A* may always grant access privileges to user *B* on a data objects which *A* owns.

T F (r) The simple security property for mandatory access control states that a user *U* has access to a data object *P* if and only if the clearance of *U* is at least as high as the clearance of *P*.

T F (s) By requiring all queries to return large numbers of tuples, it is possible to prevent statistical tracking on a database.

T F (t) A substantial limitation of object-relational approaches is that there is no way to incorporate persistent data.

Name and ID number: ____

(2: 60 points total; 6 points for each question) For each of the questions below, place an X between the parentheses to the left of the correct responses to the question. A question will have at least one correct choice, but it may have several. Mark all correct choices.

(a) In a dense secondary index:

( ) The entire index is stored in main memory.
( ) The indexed attribute is not the primary key.
( ) The records are stored sequentially.
( ) There is a distinct index entry for each record.

(b) A major advantage of a B^+-tree over a B-tree is that:

( ) The algorithm to insert a new data item is much simpler.
( ) A B^+-tree is more fully balanced.
( ) The data may be indexed on two keys at once.
( ) Sequential access to the data is supported.

(c) A situation in which extendible hashing might be a superior alternative to a B^+-tree is:

( ) The amount of fast memory available to store the indexed is limited.
( ) The data must be accessed sequentially as well as by random key.
( ) The field is often joined with a non-indexed field of another relation.
( ) The field is a secondary index.

(d) The semijoin operation:

( ) is of primary use in the efficient processing of queries in distributed systems.
( ) is logically equivalent to a projection of the join.
( ) is commutative.
( ) can result in unsafe queries if care is not taken.

(e) When joining two relations of approximately equal size, one of which has a hash index on the joining column:

( ) Building a temporary common hash table is a useful technique.
( ) Having a hash index on the joining column of the other relation as well is not of great additional advantage.
( ) Having a sequential index on the joining column of the other relation as well is not of great additional advantage.
( ) A merging algorithm is a particularly attractive choice.

(f) In a schedule of transactions which is view serializable but not conflict serializable:

( ) Deadlock is not possible.
( ) A 2PL locking schedule is not possible.
( ) There must be at least one blind write.
( ) There must be at least one blind read.

(g) The major advantage of a conservative 2PL schedule over one which is not conservative is that:

( ) A greater degree of parallelism is possible.
( ) Deadlock is not possible when conservative schedules are employed.
( ) A write-ahead log protocol is not necessary.
( ) It is impossible to have a nonrecoverable schedule when conservative 2PL is employed.

(h) The major advantage of a strict 2PL schedule over one which is not strict is that :

( ) A greater degree of parallelism is possible.
( ) Deadlock is not possible.
( ) A write-ahead log protocol is not necessary.
( ) It is impossible to have a nonrecoverable schedule when strict 2PL is employed.

(i) A schedule of transactions is said to be *nonrecoverable* in the case that:

( ) Strict 2PL is not employed.
( ) Recovery from a system crash is impossible.
( ) Recovery from deadlock is not possible.
( ) Recovery may involve rollback of committed transactions.

(j) The fundamental techniques for managing object persistence in object-oriented database systems are:

( ) naming.
( ) reachability.
( ) dynamic scope.
( ) dynamic extent.

(3: 80 points total) Important: The answers to this question must be indicated below; do not write the answers to this question on a blank sheet.

Consider the relational schema with relation $R[ABCDEFG]$ and functional dependencies $\{AB \rightarrow C, C \rightarrow DE, E \rightarrow FG, G \rightarrow BC\}$.

(a) Find all of the candidate keys for this schema. Write the answer below.

For each decomposition given below, identify those properties which the decompositions possesses. Do this by circling (drawing a ring around) the names of the properties which that decomposition has. Circle the word none just in the case that none of the other five properties applies. (Thus, it is never correct to circle nothing at all, and it is never correct to circle the word none together with something else.)

(b) The decomposition $\{R[ABC], R[BCDEFG]\}$:

none 2NF 3NF BCNF lossless dependency-preserving

(c) The decomposition $\{R[ABC], R[CDEFG]\}$:

none 2NF 3NF BCNF lossless dependency-preserving

(d) The decomposition $\{R[ABCDE], R[EFG]\}$:

none 2NF 3NF BCNF lossless dependency-preserving

(e) The decomposition $\{R[ABCG], R[CDEFG]\}$:

none 2NF 3NF BCNF lossless dependency-preserving

(f) The decomposition $\{R[ABC], R[BCG], R[CDEFG]\}$:

none 2NF 3NF BCNF lossless dependency-preserving

(g) The decomposition $\{R[AC], R[BCG], R[CDE], R[EFG]\}$:

none 2NF 3NF BCNF lossless dependency-preserving

(h) The decomposition $\{R[AB], R[BC], R[CDE], R[EFG]\}$:

none 2NF 3NF BCNF lossless dependency-preserving

(i) The decomposition $\{R[ABCDEFG]\}$:

none 2NF 3NF BCNF lossless dependency-preserving

Name and ID number: ______________________________

(4: 60 points total) A simple database system has three data objects, named x, y, and z. The following four transactions are given.

$$
\begin{aligned}
T_1 &= \mathsf{r}_1(y)\mathsf{w}_1(y)\mathsf{w}_1(x) \\
T_2 &= \mathsf{w}_2(x)\mathsf{r}_2(z)\mathsf{w}_2(z) \\
T_3 &= \mathsf{r}_3(y)\mathsf{w}_3(z)\mathsf{w}_3(y) \\
T_4 &= \mathsf{r}_4(x)\mathsf{r}_4(z)\mathsf{w}_4(x)\mathsf{w}_4(z)
\end{aligned}
$$

Shown below are three schedules for this set of transactions. For each of these schedules, do the following **on this sheet**:

- Using the nodes provided, draw the edges of the conflict graph.
- Under the name of each property (*serial*, *view serializable*, *conflict serializable*), circle **T** (*true*) or **F** (*false*), depending upon whether or not the schedule possesses that property.
- If the schedule is view serializable, give the transaction sequence (*e.g.*, $T_4T_1T_3T_2$) of an equivalent serial schedule. If the schedule is not view serializable, no answer is required for this part.

(a) $\mathsf{r}_4(x)\mathsf{w}_2(x)\mathsf{r}_4(z)\mathsf{w}_4(x)\mathsf{w}_4(z)\mathsf{r}_1(y)\mathsf{w}_1(y)\mathsf{w}_1(x)\mathsf{r}_2(z)\mathsf{w}_2(z)\mathsf{r}_3(y)\mathsf{w}_3(z)\mathsf{w}_3(y)$

T_1 T_2 serial view serializable conflict serializable

T F T F T F

T_3 T_4 Equivalent serial schedule: ______________________

(b) $\mathsf{r}_4(x)\mathsf{r}_3(y)\mathsf{w}_2(x)\mathsf{r}_1(y)\mathsf{w}_1(y)\mathsf{r}_4(z)\mathsf{r}_2(z)\mathsf{w}_2(z)\mathsf{w}_4(x)\mathsf{w}_3(z)\mathsf{w}_3(y)\mathsf{w}_4(z)\mathsf{w}_1(x)$

T_1 T_2 serial view serializable conflict serializable

T F T F T F

T_3 T_4 Equivalent serial schedule: ______________________

(c) $\mathsf{r}_3(y)\mathsf{w}_3(z)\mathsf{w}_3(y)\mathsf{r}_4(x)\mathsf{r}_4(z)\mathsf{w}_4(x)\mathsf{w}_2(x)\mathsf{w}_4(z)\mathsf{r}_2(z)\mathsf{w}_2(z)\mathsf{r}_1(y)\mathsf{w}_1(y)\mathsf{w}_1(x)$

T_1 T_2 serial view serializable conflict serializable

T F T F T F

T_3 T_4 Equivalent serial schedule: ______________________

Name and ID number: ____

(5: 60 points total) Shown below is the sequence of read and write operations of a transaction T for a system with four data objects, x, y, z, and w.

$$\mathsf{Begin}(T) \quad \mathsf{r}(x)\mathsf{w}(x)\mathsf{r}(y)\mathsf{w}(w)\mathsf{r}(z)\mathsf{w}(z) \quad \mathsf{Commit}(T)$$

The following notation is employed:

r(x)	Read data item x	w(x)	Write data item x
RL(x)	Read lock data item x	WL(x)	Write lock data item x
UL(x)	Unlock data item x	Up(x)	Upgrade to a write lock on data item x
Begin(T)	Begin transaction	Dn(x)	Downgrade to read lock on data item x
Commit(T)	Commit transaction		

For each of the augmented sequences shown below, indicate whether it obeys the *two-phase locking protocol* (2PL), the *conservative* two-phase locking protocol, or the *strict* two-phase locking protocol. Do this by circling the appropriate key words above the sequence. Note that, in a given instance, more than one of these properties, or none at all, may apply. Circle *all* applicable properties. If none of the choices applies, then circle the word *none*. **You must submit your answers on this sheet.**

(a) The sequence below is (cirle the appropriate choices): none 2PL conservative strict

Begin(T)
WL(x) r(x) w(x) UL(x) RL(y) r(y) UL(y) WL(w) w(w) UL(w) WL(z) r(z) w(z) UL(z)
Commit(T)

(b) The sequence below is (cirle the appropriate choices): none 2PL conservative strict

Begin(T)
RL(x) WL(y) WL(z) WL(x)Up(x) r(x) w(x) UL(x) r(y) w(w) UL(w) UL(y) r(z) w(z) UL(z)
Commit(T)

(c) The sequence below is (cirle the appropriate choices): none 2PL conservative strict

Begin(T)
WL(x) r(x) w(x) RL(y) r(y) UL(y) WL(w) w(w) WL(z) r(z) w(z) UL(z)UL(x) UL(w)
Commit(T)

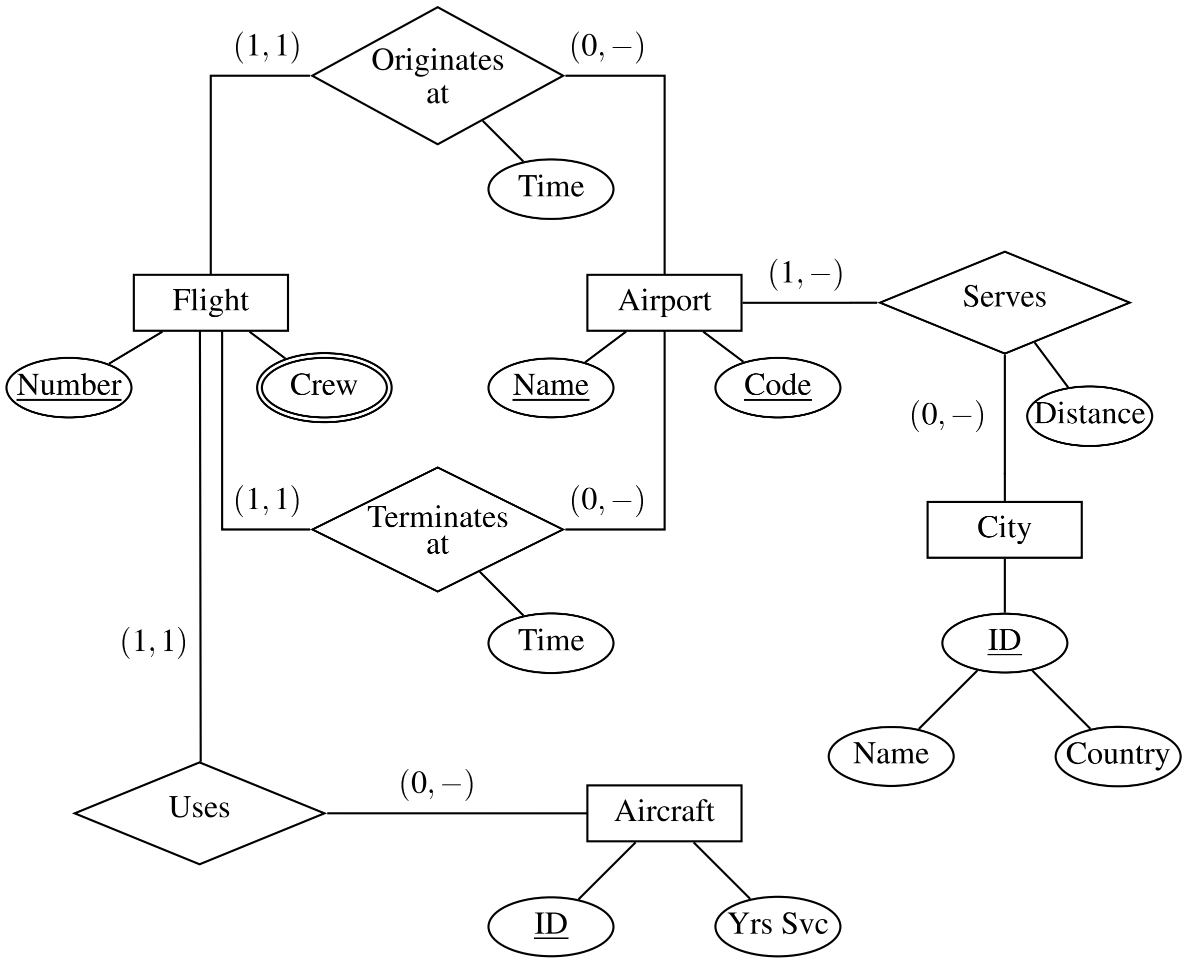
(d) The sequence below is (cirle the appropriate choices): none 2PL conservative strict

Begin(T)
WL(x) r(x) w(x) RL(y) r(y) WL(w) w(w) WL(z) r(z) w(z) UL(z)UL(x) UL(w) UL(y)
Commit(T)

(e) The sequence below is (cirle the appropriate choices): none 2PL conservative strict

Begin(T)
WL(x) RL(y) WL(w) WL(z) r(x) w(x) r(y) Up(y) w(w) r(z) w(z) UL(z) UL(y) UL(x) UL(w)
Commit(T)

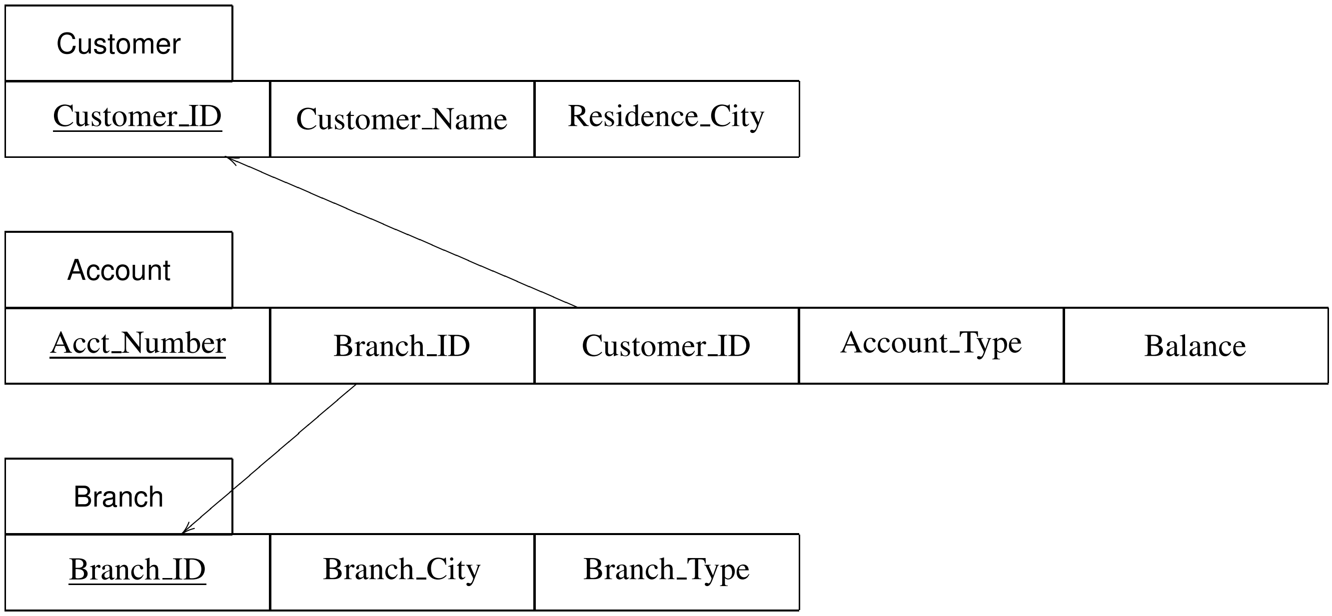
(6: 80 points) Shown below is an ER-diagram for an airline database. Using the techniques developed in the course, map this diagram into an equivalent relational schema. Show all keys, primary and foreign, and link foreign keys to their primary partners.



Important: If a primary and/or foreign key consists of more than one attribute, make sure that your notation identifies and links these composite keys as groups.

Note: The (x,y) notation gives the minimum and maximum number of times that a given instance of the entity may participate in the relationship. Thus, $(1,1)$ means exactly one, and $(0,-)$ means any number.

The following schema applies to Problems 7 through 10.



together with the following three queries, are the common context in which Problems 7, 8, and 9 below, should be solved. In each case, the primary key is underlined, and the arrows run from foreign keys to their primary counterparts.

(a) Find the names of those customers who have an account in a branch located in Vännäs, and an account in a branch located in Umeå.

(b) Find the names of those customers who have exactly one type of account.

(c) Find the names of those customers who have an account of every type which listed in the database.

7. (60 points total; 20 for each part) Solve each of the three queries (a) - (c) in the relational algebra. Functional operators, such as count and average, may not be used.

8. (60 points total; 20 for each part) Solve each of the three queries (a) - (c) in the tuple relational calculus. Functional operators, such as count and average, may not be used.

9. (60 points total; 20 for each part) Solve each of the three queries (a) - (c) using SQL. Functional operators, such as count and average, may be used in the answers to this question.

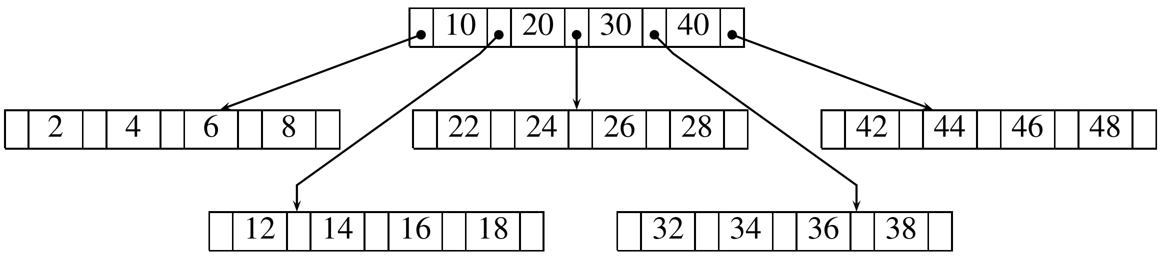
(10: 60 points total; 20 points for each part) Provide solutions, in SQL, to the following additional queries. All SQL operations may be used, including aggregation operations.

(a) For each customer with at least one account, display the customer ID, customer name, and the average balance over all accounts held by that customer. If a customer does not have any accounts, do not display anything for that customer.

(b) For each customer who has a total balance in all accounts of at least 100000, display the customer ID, the customer name, the total balance in all accounts, and the total number of accounts.

(c) Display the customer ID and and customer name of that customer who owns the account which has the greatest balance. If there is a tie, display all such customers.

(11: 60 points) Shown below is a B-tree of order four, with keys taken from the natural numbers under the standard ordering. Give the B-tree which results from insertion of a record with key 50. To obtain credit, the tree which results from application of the standard insertion algorithm must be used; it is not correct simply to provide an arbitrary B-tree with the correct key values.



(12: 80 points total) The following parameters apply to a relation which is to be stored in a B^+-tree. Assume that each node, interior or leaf, is stored in one page, and that each tuple is stored as one record.

Number of tuples in the relation	4×10^9
Size of one tuple	150 bytes
Size of pointer to a B^+-tree node	4 bytes
Size of internal B^+-tree key	16 bytes
Page size	2048 bytes
Sequential-access pointers in the leaf nodes	8 bytes total

Answer the following. To receive credit, you must show your work, including the formulas used. (If you show just numerical computations, without an underlying formula, full credit will not be awarded.)

(a: 12 points) Compute the maximum number of indices per non-leaf node.

(b: 12 points) Compute the maximum number of records per leaf node.

(c: 12 points) Compute the maximum depth that such a B^+-tree may have. (The depth is defined as the length of a path from the root to a leaf.)

(d: 12 points) Compute the minimum depth that such a B^+-tree may have.

(e: 32 points) For a B^+-tree of the minimum depth computed in (d), compute the average number of indices in the interior nodes other than the root, assuming that the root contains the maximum possible number of indices, and that the leaf nodes contain the maximum possible number of records. State whether or not this value is consistent with the constraints on the minimum and maximum number of indices in an interior node of a B^+-tree. In addition, if this value is consistent, compute the number of interior nodes and the number of leaf nodes, as well as the average number of records in each leaf node.

Appendix

A useful formula for uniform (m,r,d) B-trees:

$$R(m,r,d) = (m+1) \times (r+1)^d - 1$$

with

$$\begin{aligned}
m &= \text{the total number of indices at the root node.} \\
r &= \text{the total number of records in each non-root node.} \\
d &= \text{the depth of the tree; } \textit{i.e.}, \\
&\ \text{the length of a path from the root to a(ny) leaf node.} \\
R(m,r,d) &= \text{the total number of records in the tree.}
\end{aligned}$$

A useful formula for uniform (m,q,r,d) B$^+$-trees:

$$R(m,q,r,d) = (m+1) \times (q+1)^{d-1} \times r$$

with

$$\begin{aligned}
m &= \text{the total number of indices at the root node.} \\
q &= \text{the total number of indices in each non-root, non-leaf node.} \\
r &= \text{the total number of records in each leaf node.} \\
d &= \text{the depth of the tree; } \textit{i.e.}, \\
&\ \text{the length of a path from the root to a(ny) leaf node.} \\
R(m,q,r,d) &= \text{the total number of records in the tree.}
\end{aligned}$$