Quantum Information Review
An International Journal

http://dx.doi.org/10.12785/qir/010102

Database Manipulation Operations on Quantum Systems

Ahmed Younes[1]

[1]Department of Mathematics and Compiter Science, Faculty of Science,Alexandria University, Alexandria, Egypt

Received: 7 Nov. 2012, Revised: 22 Dec. 2012, Accepted: 26 Dec. 2012
Published online: 1 Jan. 2013

Abstract: Manipulating a database system on a quantum system is an essential aim to benefit from the promising speed-up of quantum computers over classical computers in areas that take a vast amount of storage and processing time such as in databases. In this paper, the basic operations for manipulating the data in a quantum database will be defined, e.g. INSERT, UPDATE, DELETE, SELECT, backing up and restoring a database file. This gives the ability to perform the data processing, that usually takes a long processing time on a classical database system, in a simultaneous way on a quantum system. Defining a quantum version of more advanced concepts used in database systems, e.g. the referential integrity and the relational algebra, is a normal extension to this work.

Keywords: Quantum database, quantum query, quantum circuit

1 Introduction

Quantum computers promise to do computation more powerfully [18] than classical computers due to the ability of a quantum systems to be in some states that have no equivalence in a classical computer such as a superposition of values and/or an entanglement between some particles of a quantum system [9]. A superposition is the ability of the system to be in more than one state simultaneously over the same physical space while an entanglement is the existence of a hidden correlation between the particles of a quantum system [2] so that applying an operation on an entangled particle will apply that operation on all the particles entangled with that particle [4]. A quantum system exploits a superposition to perform parallel computation on many values simultaneously at the bit level while a classical computer can perform simultaneous operations at the CPU level [15].

To extract information from a quantum system, a measurement must be used [15]. If that quantum system exists in a superposition, the measurement will break the superposition to one of the superposed values in a random manner. Otherwise, a quantum system behaves classically, i.e. if no superposition exists. Many useful methods are known to increase the probability of a required value to be found with a probability close to certainty when the measurement is applied [7,12,6,5,20].

Many quantum algorithms exploit a superposition and/or an entanglement to perform computation faster than it can be done on classical computers [17,10,23], where all the possible inputs of a problem are examined simultaneously. A superposition can be understood as a list of values superposed together on the same memory location. A database file is a two dimensional data structure (a table) where every column represents a field over certain data type and every row represents a record (a collection of related fields) [14]. A database file is simply a list of unique records. Combining the fields in each record in some fixed binary representation, a list of records can be manipulated as a list of values that can exist in a superposition on a quantum system.

Structured Query Language (SQL) is a tool widely used in manipulating the classical databases [14]. Basic operations in SQL include inserting a new record to a database file (INSERT), updating an existing record (UPDATE), deleting an exiting record (DELETE), selecting (SELECT) and performing an arbitrary operation on some records, backing up a portion of a database (BACKUP), and restoring the backup (RESTORE). This paper proposes some elementary operations for a Quantum Query Language (QQL) required to manipulate a database file exists in a superposition.

The paper is organized as follows: Section 2 briefly reviews the basic concepts in quantum computation. Section 3 defines the basic quantum transformations

* Corresponding author e-mail: ayounes2@yahoo.com

required to construct the QQL. Section 4 defines the basic operators of the QQL. Section 5 will conclude the work showing some future directions to the way of constructing a complete Quantum Database Management System (QDBMS).

2 Quantum Systems

2.1 Quantum Bits

The quantum bit (*qubit* [16]) is the quantum analogue of the classical bit. The basic difference between the qubit and the classical bit is that the qubit can exist in a linear superposition of the two states $|0\rangle$ and $|1\rangle$ at the same time (*Quantum Parallelism*),

$$a\,|0\rangle + b\,|1\rangle\,, \tag{1}$$

where a and b are complex numbers called the amplitudes of system and satisfy the condition $|a|^2 + |b|^2 = 1$. The states $|0\rangle$ and $|1\rangle$ can be taken as the classical bit values 0 and 1 respectively. $|\ \rangle$ is called the *Dirac notation* [8] and is considered as the standard notation for describing quantum states. In quantum circuits shown in this paper, a qubit will be represented as a horizontal line and the time flow of the circuit will be from left to right.

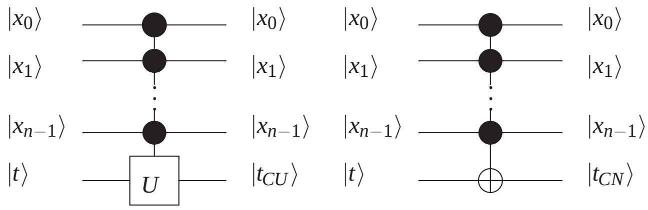
Consider the case where we have a quantum system (quantum register) with more than one qubit. In conventional computers, a two-bit register will be able to carry only one value out of the four possible values $\{00, 01, 10, 11\}$ at a time. The corresponding states in a two-qubit quantum register will be $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$, so its state in a superposition can be represented as,

$$|\psi\rangle = a_0\,|00\rangle + a_1\,|01\rangle + a_2\,|10\rangle + a_3\,|11\rangle\,, \tag{2}$$

where a_i are complex numbers satisfy the condition $\sum_i |a_i|^2 = 1$. Any measurement applied on the qubits will lead to one of the four possible states $|i\rangle$ with probability $|a_i|^2$, where i is the integer representation of that state.

2.2 Quantum Gates

In general, quantum computation process can be understood as applying a series of quantum gates followed by applying a measurement to obtain the result [13]. Quantum gates used during the computation must follow the fundamental laws of quantum physics [8]. To satisfy this condition, any matrix U that represents a quantum gate must be unitary, i.e. the inverse of that matrix must be equal to its complex conjugate transpose: $U^{-1} = U^{\dagger}$ and $UU^{\dagger} = I$, where U^{-1} denotes the inverse of U, $U^{\dagger}$ denotes the complex conjugate transpose of U and I is the identity matrix. Any gate applied on a quantum register of size n can be understood by its action on the basis vectors and can be represented as a unitary matrix of size $2^n \times 2^n$.



a. Controlled-U b. *CNOT*

Fig. 1: Controlled gates. The black circle $\bullet$ indicates the control qubits, and the symbol $\oplus$ in part (b.) indicates the target qubit.

For example, the *NOT* gate is a single input/output gate that inverts the state $|0\rangle$ to $|1\rangle$ and visa versa. Its 2×2 unitary matrix: $NOT = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$. Another important example is the Hadamard (H) gate. Its 2×2 unitary matrix: $H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$. Hadamard gate has a special importance in setting up a superposition of a quantum register. Consider a three qubits quantum register $|000\rangle$, applying Hadamard gate on each of them in parallel will set up a superposition of the 2^3 possible states. Applying any operation on that register afterward will be applied on the 2^3 states simultaneously.

Controlled operations play an important role in building up quantum circuits for any given operation [1]. The Controlled-U gate is a general controlled gate with one or more control qubit(s) as shown in (1.a). It works as follows: U is applied on the target qubit $|t\rangle$ if and only if all $|x_k\rangle$ are set to $|1\rangle$, i.e. qubits will be transformed as follows,

$$\begin{array}{l} |x_k\rangle \rightarrow |x_k\rangle\,, k: 0 \rightarrow n-1, \\ |t\rangle \ \ \rightarrow |t_{CU}\rangle = U^{x_0 x_1 \ldots x_{n-1}}\,|t\rangle\,, \end{array} \tag{3}$$

where $x_0 x_1 \ldots x_{n-1}$ in the exponent of U denotes the *AND*-ing operation of the qubit-values $x_0, x_1, \ldots, x_{n-1}$.

If U in the general case is replaced with the *NOT* gate mentioned above, the resulting gate is called *CNOT* gate (shown in (1.b). It inverts the target qubit if and only if all the control qubits are set to $|1\rangle$ as follows,

$$\begin{array}{l} |x_k\rangle \rightarrow |x_k\rangle\,; k: 0 \rightarrow n-1, \\ |t\rangle \ \ \rightarrow |t_{CN}\rangle = |t \oplus x_0 x_2 \ldots x_{n-1}\rangle\,, \end{array} \tag{4}$$

where $\oplus$ is the classical XOR operation.

3 Basic Operations

Before defining the operators of the QQL, three basic operations must be defined. Firstly, a simple way to convert the standard irreversible logic operations, e.g.

AND, OR, NOT, *etc*[3], to reversible logic operations suitable for quantum computers. This has a special importance in applying an arbitrary operation based on two or more SELECT operators. Then, a quantum oracle is used to apply a *query* on a database file exists in a superposition and gives the output of the query entangled with a temporary qubit dedicated for subspace identification purposes. Finally, an operator that acts only on a certain subspace of the system to be used in the process of backing up and restoring a portion of a quantum database.

3.1 Boolean Quantum Logic (CNOT gates)

A relational expression is an expression that has two operands connected with a relational operator from the set $\{>, \geq, <, \leq, =, \neq\}$. A relational expression evaluates either to *true* (1) or to *false* (0). A logical expression is an expression that combines two or more relational expressions with logical operators such as *AND*, *OR* and *NOT*, e.g. $(x_0\, OR\, (NOT\, x_1))$, where $x_0, x_1 \in \{0, 1\}$. These sort of logical expressions cannot be used directly as quantum logical expressions because their operations are not reversible [19]. A logical expression can be understood as a Boolean function while the relational expressions are the inputs to that Boolean function.

In building quantum circuits for Boolean functions, an extra temporary qubit will be added to the system and will be initialized to state $|0\rangle$, to hold the result of the Boolean function at the end of the computation. For clarity purposes, the *CNOT* gates will be presented as follows [11]: $CNOT\,(C|t)$ is a gate where the target qubit $|t\rangle$ is controlled by a set of qubits C such that $t \notin C$, the state of the qubit $|t\rangle$ will be flipped from $|0\rangle$ to $|1\rangle$ or from $|1\rangle$ to $|0\rangle$ if and only if all the qubits in C are set to true (state $|1\rangle$), i.e. the new state of the target qubit $|t\rangle$ will be the result of *XOR*-ing the old state of $|t\rangle$ with the *AND*-ing of the states of the control qubits. For example, consider the *CNOT* gate shown in (2), it can be represented as $CNOT\,(\{x_0, x_2\}\,|x_3)$, where $\bullet$ on a qubit means that the condition on that qubit will evaluate to true if and only if the state of that qubit is $|1\rangle$, while $\oplus$ denotes the target qubit which will be flipped if and only if all the control qubits are set to true, which means that the state of the qubit $|x_3\rangle$ will be flipped if and only if $|x_0\rangle = |x_2\rangle = |1\rangle$ with whatever value in $|x_1\rangle$; i.e. $|x_3\rangle$ will be changed according to the operation $x_3 \rightarrow x_3 \oplus x_0 x_2$. If $C = \{\}$, i.e. an empty set, then the target qubit will be flipped unconditionally (*NOT* gate).

3.2 Boolean Quantum Circuits (BQC)

A general Boolean quantum circuit U of size m (size of the circuit refers to the total number of *CNOT* gates in that circuit) over n qubit quantum system with qubits

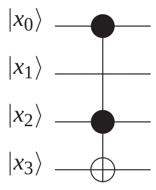


Fig. 2: $CNOT\,(\{x_0, x_2\}\,|x_3)$ gate.

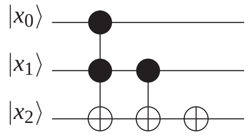


Fig. 3: Boolean quantum circuit.

$|x_0\rangle, |x_1\rangle, \ldots, |x_{n-1}\rangle$ can be represented as a sequence of *CNOT* gates [11] as follows,

$$U_g = CNOT\,(C_1|t_1)\ldots CNOT\,(C_j|t_j)\ldots CNOT\,(C_m|t_m), \quad (5)$$

where $t_j \in \{x_0, \ldots, x_{n-1}\}; C_j \subset \{x_0, \ldots, x_{n-1}\}; t_j \notin C_j$ and $j: 1 \rightarrow m$. The BQC that will be used in this paper can be represented as follows,

$$U = CNOT(C_1|t)...CNOT(C_j|t)...CNOT(C_m|t), \quad (6)$$

where $t \equiv x_{n-1}; C_j \subseteq \{x_0, \ldots, x_{n-2}\}$. For example, consider the quantum circuit shown in (3), it can be represented as follows,

$$U = CNOT(\{x_0, x_1\}|x_2).CNOT(\{x_1\}|x_2).CNOT(x_2), \quad (7)$$

Now, to trace the operations that have been applied on the target qubit $|x_2\rangle$, we will trace the operation of each of the *CNOT* gates that has been applied:

–$CNOT(\{x_0, x_1\}|x_2) \Rightarrow x_2 \rightarrow x_2 \oplus x_0 x_1$,
–$CNOT(\{x_1\}|x_2) \Rightarrow x_2 \rightarrow x_2 \oplus x_1$,
–$CNOT(x_2) \Rightarrow x_2 \rightarrow \overline{x}_2 = x_2 \oplus 1$.

Combining the three operations, we see that the complete operation applied on $|x_2\rangle$ is represented as follows,

$$x_2 \rightarrow x_2 \oplus x_0 x_1 \oplus x_1 \oplus 1. \quad (8)$$

If $|x_2\rangle$ is initialized to $|0\rangle$, applying the circuit will make $|x_2\rangle$ carry the result of the operation $(x_0 x_1 \oplus x_1 \oplus 1)$, which is equivalent to the operation $x_0 + \overline{x}_1$, i.e. $(x_0\, OR\, (NOT\, x_1))$. More details on how to convert more complex *canonical Boolean expression* (expressions use *AND*, *OR*, *NOT*) to quantum circuits using *Reed-Muller expression* (expressions use *AND*, *XOR*, *NOT*) can be found in [22].

3.3 Quantum Oracle

Consider an unstructured list L of N items. For simplicity and without loss of generality we will assume that $N = 2^n$ for some positive integer n. Suppose the items in the list are labeled with the integers $\{0,1,...,N-1\}$, and consider a Boolean function f which maps an item $i \in L$ to either 0 or 1 according to some properties this item should satisfy, i.e. $f : L \to \{0,1\}$.

It follows directly, from the discussion in the above sections, that the function f can be represented as a unitary matrix U_f. U_f will be taken as an oracle that applies a *query* on the database file and returns the results. U_f has the following effect when applied on a quantum register $|x,y\rangle$,

$$U_f : |x,y\rangle \to |x, y \oplus f(x)\rangle, \tag{9}$$

where $|x\rangle$ is a quantum register of size n and $|y\rangle$ is a temporary qubit. If $|y\rangle$ is initially set to $|0\rangle$, then U_f has the following effect on the quantum register,

$$U_f : |x,0\rangle \to |x, f(x)\rangle. \tag{10}$$

This oracle has a special importance in setting up an entanglement on the states that make the oracle evaluates to true as follows: assume that $|\psi\rangle$ is a quantum register of size $n+1$ qubits. The first n qubits in a superposition and the last qubit is an extra qubit initialized to state $|0\rangle$. Assume that U_f is a quantum oracle used to identify the states in the superposition that make f evaluate to true. Applying U_f on $|\psi\rangle$ can be understood as follows,

$$\begin{aligned} U_f|\psi\rangle &= U_f \sum_{i=0}^{2^n-1} \alpha_i |i\rangle \otimes |0\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle \otimes |f(i)\rangle \\ &= \sum_{i=0}^{2^n-1}{}' \alpha_i |i\rangle \otimes |1\rangle + \sum_{i=0}^{2^n-1}{}'' \alpha_i |i\rangle \otimes |0\rangle, \end{aligned} \tag{11}$$

where, $\sum_i{}'$ denotes a sum over i which are desired items, and $\sum_i{}''$ denotes a sum over i which are undesired items in the list, i.e. the desired items are entangled with state $|1\rangle$ of the extra qubit and the undesired items are entangled with state $|0\rangle$. So far, this can be considered as the SELECT operator since the selected states are entangled with state $|1\rangle$. Applying any operation U based on the condition that the extra qubit is in state $|1\rangle$ will be applied only of the subspace of the desired items as shown in (4). To apply an arbitrary operation U ($2^n \times 2^n$ unitary matrix) only on the subspace entangled with state $|1\rangle$, U must be transformed to a unitary matrix of size $2^{n+1} \times 2^{n+1}$ as follows,

$$U \to U \otimes |1\rangle\langle 1| + I_n \otimes |0\rangle\langle 0|, \tag{12}$$

where I_n is the identity matrix of size $2^n \times 2^n$.

3.4 Partial Diffusion

The *partial diffusion operator*, D_p, is an operator that performs amplitude alteration only on the subspace of the system entangled with the extra qubit workspace in state $|0\rangle$ [23]. The diagonal representation of D_p when applied on $n+1$ qubits system takes this form,

$$D_p = \left(H^{\otimes n} \otimes I_1\right) \left(\left(1-e^{i\varphi}\right)|0\rangle\langle 0| - I_{n+1}\right) \left(H^{\otimes n} \otimes I_1\right), \tag{13}$$

where the vector $|0\rangle$ used in Equation (13) is of length $2N = 2^{n+1}$, I_k is the identity matrix of size $2^k \times 2^k$ and φ is an arbitrary angle such that $\varphi \neq 0$. Consider a general state $|\psi\rangle$ of $n+1$ qubits register,

$$\begin{aligned} |\psi\rangle = \sum_{k=0}^{2N-1} \delta_k |k\rangle &= \sum_{j=0}^{N-1} \alpha_j (|j\rangle \otimes |0\rangle) \\ &+ \sum_{j=0}^{N-1} \beta_j (|j\rangle \otimes |1\rangle), \end{aligned} \tag{14}$$

where $\{\alpha_j = \delta_k : k \text{ even}\}$ and $\{\beta_j = \delta_k : k \text{ odd}\}$. The effect of applying D_p on $|\psi\rangle$ produces,

$$\begin{aligned} D_p|\psi\rangle &= \left(H^{\otimes n} \otimes I_1\right) \left(\left(1-e^{i\varphi}\right)|0\rangle\langle 0| - I_{n+1}\right) \left(H^{\otimes n} \otimes I_1\right)|\psi\rangle \\ &= \sum_{j=0}^{N-1} \left(1-e^{i\varphi}\right)\langle\alpha\rangle (|j\rangle \otimes |0\rangle) - \sum_{k=0}^{2N-1} \delta_k |k\rangle \\ &= \sum_{j=0}^{N-1} \left(\left(1-e^{i\varphi}\right)\langle\alpha\rangle - \alpha_j \right) (|j\rangle \otimes |0\rangle) \\ &\quad - \sum_{j=0}^{N-1} \beta_j (|j\rangle \otimes |1\rangle), \end{aligned} \tag{15}$$

where $\langle\alpha\rangle = \frac{1}{N}\sum_{j=0}^{N-1}\alpha_j$ is the mean of the amplitudes of the subspace $\alpha_j(|j\rangle \otimes |0\rangle)$, i.e. applying the operator D_p will only alter the amplitudes of the subspace $\alpha_j(|j\rangle \otimes |0\rangle)$ and will only *change the sign* of the amplitudes for the subspace $\beta_j(|j\rangle \otimes |1\rangle)$. If $\varphi = \pi$, D_p will perform the inversion about the mean only on the subspace $\alpha_j(|j\rangle \otimes |0\rangle)$ [23]. For simplicity and without loss of generality, we will use D_p with $\varphi = \pi$ throughout the rest of the paper.

4 Quantum Query Language

The architecture of the memory of the quantum system required for the operations of the QQL consists of a quantum register of size $n+t$ qubits. Initially, the system is set to state $|0\rangle^{\otimes n} \otimes |0\rangle^{\otimes t}$. The n qubits can hold up to 2^n records at a time and the t qubits will be used as temporary qubits for processing purposes. If it is required to store r records in a superposition such that $1 \leq r \leq 2^n$, then $\lceil \log_2(r) \rceil$ qubits will be used out of the n qubits.

It is important to clearly declare that the following QQL operators care only about the effects to be applied on the states of the system (values in the list). For

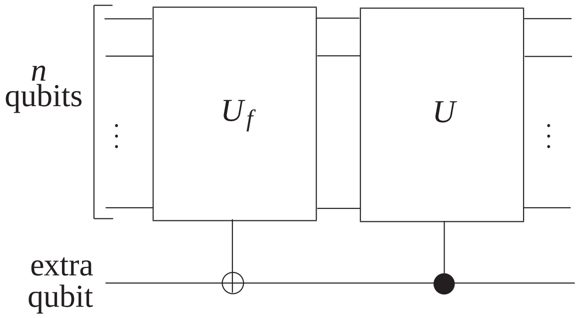


Fig. 4: Setting up entanglement on a subspace of the superposition.

simplicity, the effects to be applied on the amplitudes associated with the states in the superposition have been ignored as long as the required states exist in the superposition. The QQL operators could be associated with some quantum operators, to be constructed separately, for amplitude manipulation and to maintain the stability of the amplitudes during the processing time in specific situations.

4.1 Inserting Records to the Superposition (INSERT)

Suppose that it is required to insert some records to a superposition. To insert 2^r records directly to the superposition such that $r \leq n$, apply $H^{\otimes r} \otimes I^{\otimes n-r}$ on the first r qubits to create a system in a superposition as follows,

$$\left(\sum_{i=0}^{2^r-1} \alpha_i |i\rangle \right) \otimes |0\rangle^{\otimes n-r}. \tag{16}$$

If it is required to insert certain number of records r to a superposition such that only one record is inserted at a time, then controlled Hadamard gates can be used to achieve this goal. For example, assume that there is a quantum register of three qubits that can hold up to eight values. To insert item-by-item in sequence to the superposition, apply in sequence the set of operators $S_i, i = 0, \ldots, 7$ defined as follows (as shown in (5)),

$$\begin{aligned}
S_1 &= I \otimes I \otimes H, \\
S_2 &= I \otimes H \otimes |0\rangle\langle 0| + I \otimes I \otimes |1\rangle\langle 1|, \\
S_3 &= I \otimes I \otimes |0\rangle\langle 0| + I \otimes H \otimes |1\rangle\langle 1|, \\
S_4 &= H \otimes |00\rangle\langle 00| + \sum_{i=0, i\neq 0}^{3} |i\rangle\langle i| \otimes I, \\
S_5 &= H \otimes |01\rangle\langle 01| + I \otimes \sum_{i=0, i\neq 1}^{3} |i\rangle\langle i|, \\
S_6 &= H \otimes |10\rangle\langle 10| + I \otimes \sum_{i=0, i\neq 2}^{3} |i\rangle\langle i|, \\
S_7 &= H \otimes |11\rangle\langle 11| + I \otimes \sum_{i=0, i\neq 3}^{3} |i\rangle\langle i|.
\end{aligned} \tag{17}$$

Initially, the system is in state $|\psi_0\rangle = |000\rangle$, so, the system already contains an item. To insert the second item, apply S_1, so the system is transformed to the following,

$$|\psi_1\rangle = \alpha_0 |000\rangle + \alpha_1 |001\rangle, \tag{18}$$

where $\sum_i |\alpha_i|^2 = 1, |i\rangle \in |\psi_1\rangle$ and, to insert the third item, apply S_2 to get,

$$|\psi_2\rangle = \alpha_{00} |000\rangle + \alpha_{01} |001\rangle + \alpha_{10} |010\rangle, \tag{19}$$

and so on. If we keep applying $S_i's$ up to S_6, we get,

$$\begin{aligned}
|\psi_6\rangle = {} & \alpha_{000} |000\rangle + \alpha_{001} |001\rangle + \alpha_{010} |010\rangle \\
& + \alpha_{011} |011\rangle + \alpha_{100} |100\rangle + \alpha_{101} |101\rangle \\
& + \alpha_{110} |110\rangle.
\end{aligned} \tag{20}$$

Finally, applying S_7 will *complete* the superposition over the whole quantum register. To speed up this process a little bit, assume that it is required to insert five records to the superposition, then, firstly, apply $I \otimes H \otimes H$, to insert four records directly to the superposition in a single step, since $I \otimes H \otimes H = S_3 S_2 S_1$, then apply S_4 to insert the 5^{th} record. The natural question that might arise here is: What if it is required to insert some specific states, not necessarily in sequence, to the superposition? The answer might be more obvious after the UPDATE operator is defined in the next section.

4.2 Updating a Set of Records (UPDATE)

Updating a record is just sending the state that represents that record to another state that represents the updated record such that the record remains unique within the context of the database file. For example, assume that we have some records in a superposition as following,

$$\alpha_{000} |000\rangle + \alpha_{010} |010\rangle + \alpha_{011} |011\rangle + \alpha_{101} |101\rangle + \alpha_{110} |110\rangle. \tag{21}$$

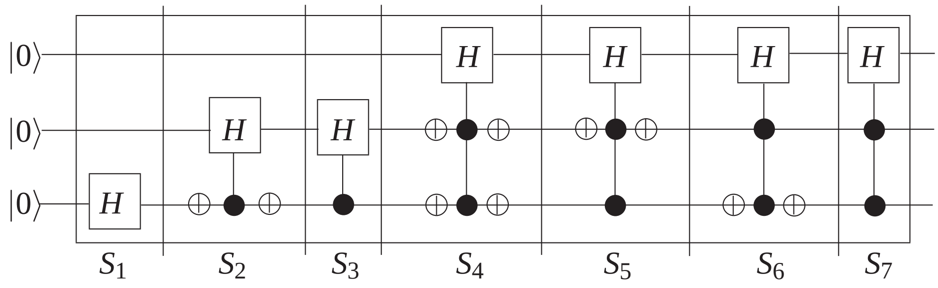


Fig. 5: Sequential insertion of items to a superposition.

To update the record $|011\rangle$ to be $|111\rangle$, i.e. it is required to transform the system shown in Equation (21) to the following system,

$$\alpha_{000}\,|000\rangle + \alpha_{010}\,|010\rangle + \alpha_{011}\,|111\rangle + \alpha_{101}\,|101\rangle + \alpha_{110}\,|110\rangle\,, \quad (22)$$

such that no change in the amplitude of the updated record, then this is a permutation. A permutation operator is a widely known operator that can be represented as a unitary matrix with 0's and 1's as its entries such that each row and column contains a single 1 and 0 everywhere else. So, the *UPDATE* operator that will transform the superposition in Equation (21) to the superposition in Equation (22) can be written as follows,

$$U_{|011\rangle\leftrightarrow|111\rangle} = \begin{bmatrix} 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&0&0&0&0&1\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&1&0\\ 0&0&0&1&0&0&0&0 \end{bmatrix}. \quad (23)$$

The UPDATE operator shown in Equation (23) is just an identity matrix of size $2^3 \times 2^3$ (3-qubit register) with the $4^{th}(|011\rangle)$ and $8^{th}(|111\rangle)$ columns been *swapped* together to affect the basis of the system as required. Notice that, applying $U_{|011\rangle\leftrightarrow|111\rangle}$ shown in Equation (23) again will *undo* the update. More update operations can be achieved using a single UPDATE operator. For example, to update the records $|000\rangle$ and $|010\rangle$ to states $|100\rangle$ and $|001\rangle$ respectively, a single UPDATE operator is required as follows,

$$U_{\substack{|000\rangle\leftrightarrow|100\rangle\\|010\rangle\leftrightarrow|001\rangle}} = \begin{bmatrix} 0&0&0&0&1&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 1&0&0&0&0&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&1 \end{bmatrix}. \quad (24)$$

A quantum circuit can be constructed for such permutation matrices using elementary *CNOT* gates [21]. We may conclude from the INSERT and UPDATE operators that any arbitrary records can be included in a superposition. They are not necessarily to be in sequence. This can be done by inserting the required number of states, then apply an UPDATE operator on some states to get the final required states in the superposition.

4.3 Deleting a Set of Records (DELETE)

Assume that we want to delete some specific records from the superposition. This problem is an interesting problem by itself. How can we remove some items from a superposition in a single step? The answer to this question is still open. In this section, we will discuss some key points that might be used to solve this problem. Firstly, we need to identify the items to be removed from the superposition. Assume that we have a Boolean function f that evaluates to true for the items we want to delete. Applying a quantum oracle U_f on the superposition taking a temporary qubit as the target qubit will identify these items by entangling the subspace of the items we want to keep in the superposition with state $|0\rangle$ of the temporary qubit, and the subspace of the items we want to delete with state $|1\rangle$ of the temporary qubit. The rest is a matter of amplitude amplification to find the temporary qubit in state $|0\rangle$ when a partial measurement is applied on that particular temporary qubit. This will erase the unnecessary states directly from the system, and will leave a superposition with the rest of the states.

4.4 Performing Conditional Operations on Some Selected Records

A usual scenario in the processing of a database is to select certain sets of records, each set is selected based on some condition, then apply an operation on the *intersection* of the selected set of records according some global condition. For example, assume that R_1 and R_2 are two selected set of records according to the two conditions c_1 and c_2 respectively. Assume that an operation U should be applied on the intersection of the selected records according to the global condition $(c_1\,AND\,(NOT\,c_2))$. (6) shows such construction where the set R_1 of records is selected by a Boolean function f_1 and the set R_2 of records is selected by a Boolean function f_2. Both selected records are combined using the global condition $(c_1\,AND\,(NOT\,c_2))$ on the last temporary qubit and a conditional application of U is done for only the records that *satisfy* the global condition. In general, to apply such an arbitrary operator U on k selected sets of records, $k+1$ temporary qubits are required.

4.5 Backing Up a Required Portion of a Database File (BACKUP)

Suppose that a copy of some states in a superposition should be stored in a safe to be protected from any arbitrary operations to be done by mistake on the superposition. To achieve this, assume that f is a Boolean function that identifies the records to be backed up. Firstly, apply U_f on the superposition taking a temporary qubit as the target qubit, this creates an entanglement between the required subspace and the temporary qubit in state $|1\rangle$, and the rest of the system entangled with the temporary qubit in state $|0\rangle$. This temporary qubit will be considered as the *key* of the safe (the safe key).

Now, there are two separate subspaces in the superposition. A subspace entangled with the temporary qubit in state $|1\rangle$ representing the items sent to the backup and the rest of the superposition that does not contain the states in the backup, entangled with state $|0\rangle$ of the temporary qubit. To create a copy of the states in the backup and insert them in the subspace entangled with state $|0\rangle$, apply the partial diffusion operator D_p on the system including the temporary qubit. The mechanism of these operations can be understood as follows: Assume that the system is initially as follows,

$$|\psi_0\rangle \sum_{i=0}^{2^n-1} \alpha_i\,|i\rangle\otimes|0\rangle. \tag{25}$$

1-*Applying the Oracle*. Apply the oracle U_f that maps the items in the list to either 0 or 1 simultaneously and stores the result in the temporary qubit:

$$\begin{aligned}|\psi_1\rangle &= U_f\,|\psi_0\rangle\\ &= U_f \sum_{i=0}^{2^n-1} \alpha_i\,|i\rangle\otimes|0\rangle = \sum_{i=0}^{2^n-1} \alpha_i\,|i\rangle\otimes|f(i)\rangle.\end{aligned} \tag{26}$$

2-*Partial Diffusion*. Let M be the number of matches, which make the oracle U_f evaluate to true, i.e. items to be sent to the backup and $N=2^n$. Assume that $\sum_i{}'$ denotes a sum over i representing the items to be sent to the backup, and $\sum_i{}''$ denotes a sum over i representing the rest of the items in the list. So, the system $|\psi_1\rangle$ shown in Equation (26) can be written as follows:

$$|\psi_1\rangle = \sum_{i=0}^{N-1}{}''\alpha_i\,(|i\rangle\otimes|0\rangle) + \sum_{i=0}^{N-1}{}'\alpha_i\,(|i\rangle\otimes|1\rangle). \tag{27}$$

Applying D_p on $|\psi_1\rangle$ will result in a new system described as follows:

$$\begin{aligned}|\psi_2\rangle = &\sum_{i=0}^{N-1}{}''a_i\,(|i\rangle\otimes|0\rangle) + \sum_{i=0}^{N-1}{}'b_i\,(|i\rangle\otimes|0\rangle)\\ &+ \sum_{i=0}^{N-1}{}'c_i\,(|i\rangle\otimes|1\rangle),\end{aligned} \tag{28}$$

where the mean used in the definition of partial diffusion operator is,

$$\langle\alpha\rangle = \frac{1}{N}\left(\sum_{i=0}^{N-1}{}''\alpha_i\right), \tag{29}$$

and a_i, b_i and c_i used in Equation (28) are calculated as follows:

$$a_i = 2\,\langle\alpha\rangle - \alpha_i,\quad b_i = 2\,\langle\alpha\rangle\,,\quad c_i = -\alpha_i. \tag{30}$$

Notice that, the states with amplitude b_i had amplitude *zero* before applying D_p. The system ends up with a copy of the required states, previously sent to the backup by the oracle, in the subspace entangled with state $|0\rangle$ of the safe key qubit. Applying any further operations on the records of the database *should* be applied by controlling that operations by the temporary qubit to be in state $|0\rangle$, in an equivalent manner to that shown in Equation (12), keeping the backup in the safe entangled with state $|1\rangle$ of the temporary qubit. Notice that, a superposition of the database file together with its backup cost an extra qubit added to the system.

4.6 Restoring a Backup

Suppose that some required records are lost from the superposition due to some invalid update and/or mistaken deletion providing that, a copy of these states has been kept in a backup and all applied operations were controlled with the safe key qubit to be in state $|0\rangle$. So, the system can be represented as follows,

$$\begin{aligned}\left|\psi'\right\rangle = &\sum_{i=0}^{N-1}{}''a_i'\,(|i\rangle\otimes|0\rangle) + \sum_{i=0}^{N-1}{}'''b_i'\,(|i\rangle\otimes|0\rangle)\\ &+ \sum_{i=0}^{N-1}{}'c_i\,(|i\rangle\otimes|1\rangle),\end{aligned} \tag{31}$$

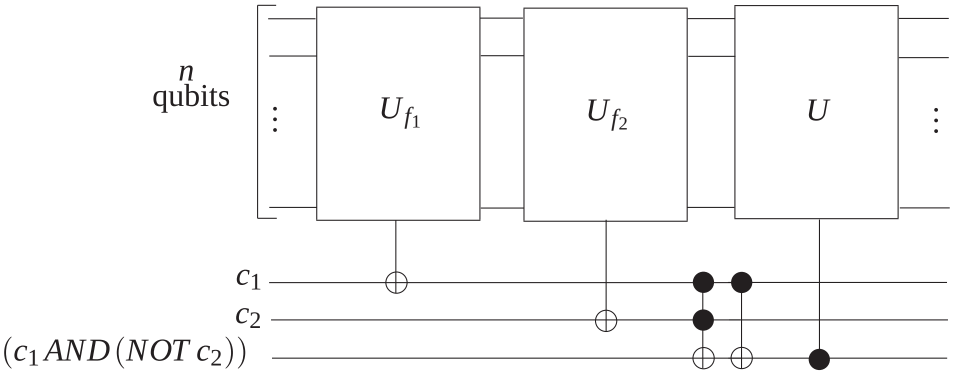


Fig. 6: Conditional application of an arbitrary operation U based on two SELECT operators, where $c_1c_2 \oplus c_1 \equiv c_1\,AND\,(NOT\,c_2)$.

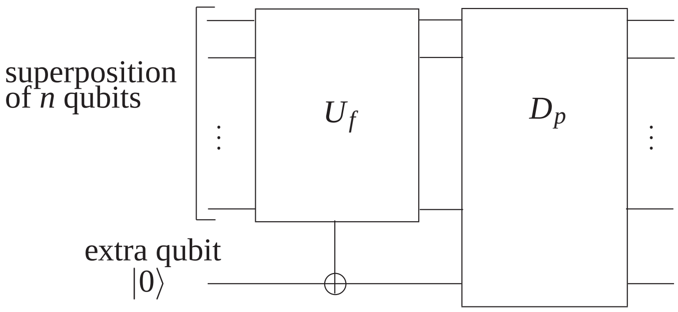


Fig. 7: Backing up a portion of a database file.

where $\sum_i{}'$ denotes a sum over i representing the items in the safe, and $\sum_i{}''$ denotes a sum over i representing the rest of the items in the list, and $\sum_i{}'''$ denotes a sum over i representing the set of the correct items left in the superposition after applying the invalid operations. Applying the oracle U_f, originally used to create the backup, on $\left|\psi'\right\rangle$ will *flip* the safe key qubit only for the items in $\sum_i{}'$ and $\sum_i{}'''$, sending the remaining correct items left in the superposition to the backup safe and restoring the items in the safe to the superposition entangled with state $|0\rangle$ as follows,

$$U_f\left|\psi'\right\rangle = \sum_{i=0}^{N-1}{}'' a_i'(|i\rangle \otimes |0\rangle) + \sum_{i=0}^{N-1}{}' c_i(|i\rangle \otimes |0\rangle) + \sum_{i=0}^{N-1}{}''' b_i'(|i\rangle \otimes |1\rangle). \quad (32)$$

Since the items in the backup safe is no longer valid (as a set of items), they can be deleted by the DELETE operator. A new fresh backup could be created using the BACKUP operator.

5 Conclusion

The quantum databases are expected to replace the classical databases once quantum computers are implemented on the commercial scale. Quantum computers can behave classically if superposition and/or entanglement are not used. Superposed quantum database will be useful in reducing the processing time where many operations could be done simultaneously on a database file as well as saving memory space. Extracting useful information from a quantum system in a superposition is still under investigation by many researchers. Distributed processing of databases could be possible where teleportation might help in sending a quantum database file in a superposition from one place to another instantly for further processing and extracting useful information.

The QQL operators defined in this paper still require further investigation to adjust the amplitudes of the system as required. General purpose amplitude manipulation techniques must be found to be combined with the operators of the QQL. Finding a quantum version of referential integrity and relational algebra to get useful information from larger databases where many database files are used could be the next research step. For a full functioning QDBMS, the QDBMS should keep track of which items are inserted for not inserting an item more than once or to delete non existence item.

To summarize, in this paper, a method for inserting exponential number of items simultaneously as well as inserting item-by-item to a superposition has been defined. A method to update many records simultaneously has been shown. A way to delete certain records from the database simultaneously has been suggested which still need special attention as a separate problem. Performing the selection of some records and applying conditional operations on the intersection of these selected records has been shown. And finally a method to backup and restore a database file without the need of vast extra memory has been proposed.

References

[1] A. Barenco, C. Bennett, R. Cleve, D. P. Divincenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin, and H. Weinfurter. Elementary gates for quantum computation. *Phys. Rev. A*, 52(5): 3457–3467, 1995.

[2] J. Bell. On the problem of hidden variables in quantum mechanics. *Reviews of Modern Physics*, 38(3): 447, 1966.

[3] C. Bennett. Logical reversibility of computation. *IBM Journal of Research and Development*, 17(6): 525–532, 1973.

[4] C. Bennett, G. Brassard, C. Crepeau, R. Jozsa, A. Peres, and W. Wootters. Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels. *Physical Review Letters*, 70: 1895–1899, 1993.

[5] M. Boyer, G. Brassard, P. Høyer, and A. Tapp. Tight bounds on quantum searching. *Fortschritte der Physik*, 46: 493, 1998.

[6] G. Brassard, P. Høyer, M. Mosca, , and A. Tapp. Quantum amplitude amplification and estimation. *arXiv e-Print quant-ph/0005055*, 2000.

[7] G. Brassard, P. Høyer, and A. Tapp. Quantum counting. *arXiv e-Print quant-ph/9805082*, 1998.

[8] P. Dirac. *The Principles of Quantum Mechanics*. Clarendon Press, Oxford, UK, 1947.

[9] R. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21:467–488, 1982.

[10] L. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing*, pages 212–219, 1996.

[11] K. Iwama, Y. Kambayashi, and S. Yamashita. Transformation rules for designing CNOT–based quantum circuits. In *Proceedings of the 39th Conference on Design Automation*, pages 419–424. ACM Press, 2002.

[12] M. Mosca. Quantum searching, counting and amplitude amplification by eigenvector analysis. In *Proceedings of Randomized Algorithms, Workshop of Mathematical Foundations of Computer Science*, pages 90–100, 1998.

[13] M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, United Kingdom, 2000.

[14] R. R. Elmasri and S. B. Navathe. *Fundamentals of Database Systems*. Addison Wesley, Boston, MA, USA, 2006.

[15] E. Rieffel and W. Polak. An introduction to quantum computing for non-physicists. *ACM Computing Surveys*, 32(3):300–335, 2000.

[16] B. Schumacher. Quantum coding. *Physical Review A*, 51:2738–2747, 1995.

[17] P. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *Proceedingsof the 35th Annual Symposium on Foundations of Computer Science*, pages 124–134. IEEE Computer Society Press, 1994.

[18] D. Simon. On the power of quantum computation. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 116–123, 1994.

[19] T. Toffoli. Reversible computing. In W. de Bakker and J. van Leeuwen, editors, *Automata, Languages and Programming*, page 632. Springer, New York, 1980. Technical Memo MIT/LCS/TM-151, MIT Lab for Computer Science (unpublished).

[20] A. Younes. Towards More Reliable Fixed Phase Quantum Search Algorithm. *Applied Mathematics and Information Sciences*, 7(1):93–98, 2013.

[21] A. Younes and J. Miller. Automated method for building CNOT based quantum circuits for Boolean functions. Technical Report CSR-03-3, University of Birmingham, School of Computer Science, arXiv e-Print quant-ph/0305134, April 2003.

[22] A. Younes and J. Miller. Representation of Boolean quantum circuits as Reed-Muller expansions. *International Journal of Electronics*, 91(7):431–444, 2004.

[23] A. Younes, J. Rowe, and J. Miller. Quantum search algorithm with more reliable behaviour using partial diffusion. In *Proceedings of the 7th International Conference on Quantum Communication, Measurement and Computing*, pages 171–174, 2004.

Ahmed Younes is presently employed as Computer Science Assistant Professor at Alexandria University. He obtained his PhD from Birmingham University (UK). He introduced a new technique, now called as 'Partial Diffusion Operator' in the field of amplitude amplification. He published papers in Quantum Algorithms, Quantum Cryptography and Reversible Circuits.