

รายการอ้างอิง

ภาษาไทย

- คณะกรรมการบัญญัติศัพท์คอมพิวเตอร์ ราชบัณฑิตยสถาน. ศัพท์คอมพิวเตอร์. กรุงเทพมหานคร: บริษัท เพื่อนพิมพ์ จำกัด, 2533
- ชิตชนก เหลือสินทรัพย์, ดร. ภาษา C และเปลือกแบบ C. กรุงเทพมหานคร: บริษัทซีเอ็ดยูเคชั่น จำกัด, 2535
- ดวงแก้ว สวามิภักดิ์, ดร. การโปรแกรมภาษา C. กรุงเทพมหานคร: บริษัทซีเอ็ดยูเคชั่น จำกัด, 2531
- ดวงแก้ว สวามิภักดิ์, ดร. ระบบฐานข้อมูล (Database System). กรุงเทพมหานคร: บริษัทซีเอ็ดยูเคชั่น จำกัด, 2534
- ไพรัตน์ สังข์สกุล, การโปรแกรมภาษา C. กรุงเทพมหานคร: โรงพิมพ์ควอลิตี้พริ้นท์, 2529
- ยรรยง เต็งอำนาจ, ดร. ระบบปฏิบัติการ (Operating System). กรุงเทพมหานคร: บริษัทซีเอ็ดยูเคชั่น จำกัด, 2533
- เย็น ภู่วรรณ, ดร. แอดวานซ์ดีเบส. กรุงเทพมหานคร: บริษัทซีเอ็ดยูเคชั่น จำกัด, 2532
- วิทยา เรื่องพรวิสุทธิ, ดร. คู่มือโปรแกรมภาษา C สำหรับผู้เริ่มต้น. กรุงเทพมหานคร: บริษัทซีเอ็ดยูเคชั่น จำกัด, 2537
- อิว ไอยรากาญจนกุล. การเขียนคอมไพเลอร์สำหรับ IBM PC. กรุงเทพมหานคร: บริษัทซีเอ็ดยูเคชั่น จำกัด, 2534

ภาษาอังกฤษ

- Bach and Maurice J. The Design Of The UNIX Operating System. New Jersey: Prentice-Hall International Inc, 1986.
- Barrett and Wagner. C and UNIX Tools for Software Design. New York: John Wiley & Sons Inc, 1996
- Ceri,Stefano and Pelagatti, Giuseppe. Distributed Databases Principles & System. New York: McGraw-Hill Book Co, 1985
- Date, C.J. An Introduction to Database System Volume I. 4th. ed. Massachusetts: Addison-Wesley Pub. Co, 1986
- Deitel, H.M. An Introduction to Operating System. Massachusetts: Addison-Wesley Pub. Co, 1983
- Hughes, David J. A lightweight Database Engine. [Machine readable data file] Fiddich Technologies, 1995
- Levine, Mason and Brown. Lex&Yacc. Sebastopol: O'Reilly&Associates Inc, 1992
- Padovano, Michael. Networking applications on UNIX System V release 4. New Jersey: Prentice-Hall Inc, 1993
- Rieken and Weiman. Adventures In UNIX Network Applications Programming. New York: John Wiley & Sons Inc, 1992
- Stevens, Al. C Database Development.. Singapore: McGraw-Hill Book Co, 1987
- Stevens, Richard. Advanced Programming in The UNIX Environment.. Massachusetts: Addison-Wesley Pub. Co, 1992
- Tremblay and Sorenson. An Introduction to Data Structures with Applications. Singapore: McGraw-Hill Book Co, 1984
- Valley, John. UNIX programmer's Reference. QUE Co, 1991
- Wiederhold, Gio. Database Design. Singapore: McGraw-Hill Book Co, 1983
- Wiederhold, Gio. File Organization for Database Design. Singapore: McGraw-Hill Book Co, 1988

ภาคผนวก

ภาคผนวก ก

ผลลัพธ์ที่ได้จากระบบในตอนทดสอบ

ในภาคผนวกนี้จะเป็นผลลัพธ์ที่ได้จากระบบในตอนทดสอบซึ่งสามารถแบ่งออกเป็น 3 ส่วน คือ

- ผลลัพธ์ที่ได้จากระบบในตอนทดสอบความสามารถทางด้านภาษาสอบถามเชิงโครงสร้าง
- ผลลัพธ์ที่ได้จากการทดสอบโปรแกรมแปลงผันจากเพิ่มข้อมูล-dBASE
- ผลลัพธ์ที่ได้จากระบบในตอนทดสอบหาค่าเฉลี่ยปฏิบัติการต่อวินาที

ส่วนที่ 1. ผลลัพธ์ที่ได้จากระบบในตอนทดสอบความสามารถทางด้านภาษาสอบถามเชิงโครงสร้าง

สามารถแบ่งตามจุดมุ่งหมายการทดสอบได้ดังนี้

1. ความสามารถในการสร้างตาราง

Welcome SQL Prototype. V.1.0.0.

Type \h for help.

SQL > create table employee (

emp_no	int primary key,
emp_name	char(25),
date_hired	char(8),
salary	real,
dept_no	int)\g

Query OK.


```
SQL > create table department (
            dept_no      int primary key,
            dept_name    char(25))\g
```

Query OK.

```
SQL > create table projects (
            proj_no      int primary key,
            proj_name    char(25),
            start_date   char(8),
            completion_date char(8),
            labor_budget int,
            dept_no      int)\g
```

Query OK.

```
SQL > create table assignment (
            emp_no      int not null,
            proj_no     int not null,
            hours_charged int)\g
```

Query

2. ความสามารถในการเพิ่มข้อมูล

Welcome SQL Prototype. V.1.0.0.

Type \h for help.

```
SQL > insert into employee
            (emp_no, emp_name, date_hired, salary, dept_no)
            values
            (1, 'jones', '06051984', 20000, 30)\g
```

Query OK

SQL > insert into employee

(emp_no, emp_name, date_hired, salary, dept_no)

values

(2, 'smith', '04021983', 22000, 20)\g

Query OK.

SQL > insert into employee

(emp_no, emp_name, date_hired, salary, dept_no)

values

(3, 'brown', '21091982', 21000, 30)\g

Query OK.

SQL > insert into employee

(emp_no, emp_name, date_hired, salary, dept_no)

values

(4, 'green', '12071920', 18000, 40)\g

Query OK.

SQL > insert into employee

(emp_no, emp_name, date_hired, salary, dept_no)

values

(5, 'white', '11111985', 25000, 20)\g

Query OK.

SQL > insert into department

(dept_no, dept_name)

values

(10, 'Personal')\g

Query OK.

```
SQL > insert into department
      (dept_no, dept_name)
      values
      (20, 'Sales')\g
```

Query OK.

```
SQL > insert into department
      (dept_no, dept_name)
      values
      (30, 'Accounting')\g
```

Query OK.

```
SQL > insert into department
      (dept_no, dept_name)
      values
      (40, 'engineering')\g
```

Query OK.

```
SQL > insert into department
      (dept_no, dept_name)
      values
      (50, 'manufacturing')\g
```

Query OK.

```
SQL > insert into projects
      (proj_no, proj_name, start_date, completion_date, labor_budget, dept_no)
      values
      (123, 'Payroll', '30011985', '12121985', 50000, 30)\g
```

Query OK.

SQL > insert into projects

(proj_no, proj_name, start_date, completion_date, labor_budget, dept_no)

values

(124, 'Accts Receivable', '01011984', '11111985', 22500, 30)\g

Query OK.

SQL > insert into projects

(proj_no, proj_name, start_date, completion_date, labor_budget, dept_no)

values

(125, 'General Ledger', '01061984', '24121985', 5000, 30)\g

Query OK.

SQL > insert into assignment

(emp_no, proj_no, hours_charged)

values

(2, 123, 120)\g

Query OK.

SQL > insert into assignment

(emp_no, proj_no, hours_charged)

values

(4, 123, 176)\g

Query OK.

SQL > insert into assignment

(emp_no, proj_no, hours_charged)

values

(1, 124, 250)\g

Query OK.

```
SQL > insert into assignment
      (emp_no, proj_no, hours_charged)
      values
      (2, 124, 432)\g
```

Query OK.

```
SQL > insert into assignment
      (emp_no, proj_no, hours_charged)
      values
      (4, 125, 640)\g
```

Query OK

3. ความสามารถในการป้องกันการเพิ่มข้อมูลลงไปตารางที่บางสดมภ์จำเป็นต้องมีข้อมูลเสมอ(not null)

Welcome SQL Prototype. V.1.0.0.

Type \h for help.

```
SQL > insert into assignment
      (emp_no, hours_charged)
      values
      (1, 250)\g
```

ERROR : Field "proj_no" cannot be null

4. ความสามารถในการป้องกันการเพิ่มข้อมูลลงไปตารางที่มีกุญแจหลักซ้ำกันไม่ได้ (unique key)

Welcome SQL Prototype. V.1.0.0.

Type \h for help.

```
SQL > insert into department
      (dept_no, dept_name)
      values
      (10, 'Finance')g

ERROR : Non unique key value in field 'dept_no'
```

5. ความสามารถในการสอบถามข้อมูลในรูปของ wildcard และมีกำหนดเงื่อนไข
แสดงข้อมูลใน Where Clause

Welcome SQL Prototype. V.1.0.0.

Type \h for help.

```
SQL > select *
      from employee\g
```

Query OK.

5 rows matched.

emp_no	emp_name	date_hired	salary	dept_no
1	jones	06051984	20000	30
2	smith	04021983	22000	20
3	brown	21091982	21000	30
4	green	12071920	18000	40
5	white	11111985	25000	20

```
SQL > select *
      from employee
      where emp_no = 4\g
```

Query OK.

1 rows matched.

emp_no	emp_name	date_hired	salary	dept_no
4	green	12071920	18000	40

```
SQL > select *
      from employee
      where emp_name = 'green'\g
```

Query OK.

1 rows matched.

emp_no	emp_name	date_hired	salary	dept_no
4	green	12071920	18000	40

```
SQL > select *
      from employee
      where emp_name like '%ow%'\g
```

Query OK.

1 rows matched.

emp_no	emp_name	date_hired	salary	dept_no
3	brown	21091982	21000	30

6. ความสามารถในการแก้ไขข้อมูล

Welcome SQL Prototype. V.1.0.0.

Type \h for help.

```
SQL > select *  
      from employee\g
```

Query OK.

5 rows matched.

```
+-----+-----+-----+-----+-----+  
| emp_no | emp_name      | date_hired | salary | dept_no |  
+-----+-----+-----+-----+-----+  
| 1      | jones         | 06051984  | 20000  | 30      |  
| 2      | smith        | 04021983  | 22000  | 20      |  
| 3      | brown        | 21091982  | 21000  | 30      |  
| 4      | green        | 12071920  | 18000  | 40      |  
| 5      | white        | 11111985  | 25000  | 20      |  
+-----+-----+-----+-----+-----+
```

```
SQL > update employee  
      set salary = 30000  
      where emp_name = 'green'\g
```

Query OK

```
SQL > select *  
      from employee\g
```

Query OK.

5 rows matched.

emp_no	emp_name	date_hired	salary	dept_no
1	jones	06051984	20000	30
2	smith	04021983	22000	20
3	brown	21091982	21000	30
4	green	12071920	30000	40
5	white	11111985	25000	20

7. ความสามารถในการสอบถามข้อมูลจาก 2 ตารางที่สัมพันธ์กัน

Welcome SQL Prototype. V.1.0.0.

Type \h for help.

SQL > select *

from employee\g

Query OK.

5 rows matched.

emp_no	emp_name	date_hired	salary	dept_no
1	jones	06051984	20000	30
2	smith	04021983	22000	20
3	brown	21091982	21000	30
4	green	12071920	30000	40
5	white	11111985	25000	20

```
SQL > select *
      from department\g
```

Query OK.

5 rows matched.

dept_no	dept_name
10	Personal
20	Sales
30	Accounting
40	engineering
50	manufacturing

```
SQL > select a.emp_no, a.emp_name, b.dept_name
      from employee a, department b
      where a.dept_no = b.dept_no\g
```

Query OK.

5 rows matched.

emp_no	emp_name	dept_name
1	jones	Accounting
2	smith	Sales
3	brown	Accounting
4	green	engineering
5	white	Sales

8. ความสามารถในการสอบถามข้อมูล โดยมีการแสดงผลเรียงตามลำดับตาม
 สดมภ์ที่กำหนดไว้ใน Order clause

Welcome SQL Prototype. V.1.0.0.

Type \h for help.

```
SQL > select a.emp_no, a.emp_name, b.dept_name
        from employee a, department b
        where a.dept_no = b.dept_no\g
```

Query OK.

5 rows matched.

emp_no	emp_name	dept_name
1	jones	Accounting
2	smith	Sales
3	brown	Accounting
4	green	engineering
5	white	Sales

```
SQL > select a.emp_no, a.emp_name, b.dept_name
        from employee a, department b
        where a.dept_no = b.dept_no
        order by a.emp_name\g
```

Query OK.

5 rows matched.

emp_no	emp_name	dept_name
3	brown	Accounting
4	green	engineering
1	jones	Accounting
2	smith	Sales
5	white	Sales

9. ความสามารถในการสอบถามข้อมูล โดยมีการแสดงผลข้อมูลที่ซ้ำกันเพียงข้อมูลเดียว

Welcome SQL Prototype. V.1.0.0.

Type \h for help.

```
SQL > select emp_name, date_hired, salary, dept_no
      from employee
      order by emp_name\g
```

Query OK.

5 rows matched.

emp_name	date_hired	salary	dept_no
brown	21091982	21000	30
green	12071920	30000	40
jones	06051984	20000	30
smith	04021983	22000	20
white	11111985	25000	20

SQL > insert into employee

(emp_no, emp_name, date_hired, salary, dept_no)

values

(6, 'jones', '06051984', 20000, 30)\g

Query OK.

SQL > select emp_name, date_hired, salary, dept_no

from employee

order by emp_name\g

Query OK.

6 rows matched.

emp_name	date_hired	salary	dept_no
brown	21091982	21000	30
green	12071920	30000	40
jones	06051984	20000	30
jones	06051984	20000	30
smith	04021983	22000	20
white	11111985	25000	20

```
SQL > select distinct emp_name, date_hired, salary, dept_no
      from employee
      order by emp_name\g
```

Query OK.

5 rows matched.

emp_name	date_hired	salary	dept_no
brown	21091982	21000	30
green	12071920	30000	40
jones	06051984	20000	30
smith	04021983	22000	20
white	11111985	25000	20

10. ความสามารถในการลบข้อมูล

Welcome SQL Prototype. V.1.0.0.

Type \h for help.

```
SQL > select *
        from employee\g
```

Query OK.

6 rows matched.

emp_no	emp_name	date_hired	salary	dept_no
1	jones	06051984	20000	30
2	smith	04021983	22000	20
3	brown	21091982	21000	30
4	green	12071920	30000	40
5	white	11111985	25000	20
6	jones	06051984	20000	30

```
SQL delete
        from employee
        where emp_no = 6\g
```

Query OK.

```
SQL > select *
        from employee\g
```

Query OK.

5 rows matched.

emp_no	emp_name	date_hired	salary	dept_no
1	jones	06051984	20000	30
2	smith	04021983	22000	20
3	brown	21091982	21000	30
4	green	12071920	30000	40
5	white	11111985	25000	20

11. ความสามารถในการสอบถามข้อมูลมากกว่า 2 ตารางที่สัมพันธ์กัน

Welcome SQL Prototype. V.1.0.0.

Type \h for help.

```
SQL > select *
      from employee\g
```

Query OK.

5 rows matched.

emp_no	emp_name	date_hired	salary	dept_no
1	jones	06051984	20000	30
2	smith	04021983	22000	20
3	brown	21091982	21000	30
4	green	12071920	30000	40
5	white	11111985	25000	20


```
SQL> select *  
      from department\g
```

Query OK.

5 rows matched.

```
+-----+-----+  
| dept_no | dept_name |  
+-----+-----+  
| 10      | Personal  |  
| 20      | Sales     |  
| 30      | Accounting|  
| 40      | engineering|  
| 50      | manufacturing|  
+-----+-----+
```

```
SQL > select *  
      from project\g
```

ERROR : Error reading table "project" definition

```
SQL > select *  
      from assignment\g
```

Query OK.

5 rows matched.

emp_no	proj_no	hours_charged
2	123	120
4	123	176
1	124	250
2	124	432
4	125	640

```
SQL > select a.emp_name, c.proj_name, b.dept_name
      from employee a, department b, projects c, assignment d
      where a.emp_no = d.emp_no
            and c.proj_no = d.proj_no
            and a.dept_no = b.dept_no
      order by a.emp_name\g
```

Query OK.

5 rows matched.

emp_name	proj_name	dept_name
green	Payroll	engineering
green	General Ledger	engineering
jones	Accts Receivable	Accounting
smith	Accts Receivable	Sales
smith	Payroll	Sales

ส่วนที่ 2. ผลลัพธ์ที่ได้จากการทดสอบโปรแกรมแปลงผันจากเพิ่มข้อมูล-dBASE

Script started on Thu Apr 25 18:12:20 1996

phueng: {1} db2sql -l -p emp_no -d test -t employee -c -vvv employee.dbf

Opening dbase file

dbf-file: employee.dbf, database: test, table: employee

Number of records: 5

NAME: LENGTH: TYPE:

NAME:	LENGTH:	TYPE:
EMP_NO	5	N
EMP_NAME	25	C
DATE_HIRED	8	D
SALARY	10	N
DEPT_NO	5	N

Making connection to server

Selecting database

Dropping table (if exists)

Building CREATE-clause

Sending create-clause

CREATE TABLE employee (emp_no int primary key,emp_name char (25),date_hired char (8),salary real,dept_no int)

Inserting records

Inserting record 0

Closing up....

phueng: {2} sql test

Welcome SQL Prototype. V.1.0.0.

Type \h for help.

SQL > employee\f

Database = test

Table = employee

Field	Type	Length	Not Null	Key
emp_no	int	4	Y	Y
emp_name	char	25	N	N
date_hired	char	8	N	N
salary	real	8	N	N
dept_no	int	4	N	N

SQL > select * from employee\g

Query OK.

5 rows matched.

emp_no	emp_name	date_hired	salary	dept_no
1	jones	19840506	20000	30
2	smith	19830204	22000	20
3	brown	19820921	21000	30
4	green	19200712	18000	40
5	white	19851111	25000	20

phueng: {3} exit

Script done on Thu Apr 25 18:14:22 1996

ส่วนที่ 3. ผลลัพธ์ที่ได้จากระบบในตอนทดสอบหาค่าเฉลี่ยปฏิบัติการต่อวินาที

1. ทดสอบเพิ่มข้อมูลและสอบถามข้อมูล 100 ครั้ง

Test SQL Prototype Performance. Test machine = FreeBSD phueng.ac.th 2.0.5-

RELEASE FreeBSD 2.0.5-RELEASE #0: Sat Jun 10 10:46:56 1995

jkh@westhill.cdrom.com:/usr/src/sys/compile/GENERIC i386

Insert 100 rows into new key table gave :-

0.56 seconds real time

Total time = .56

Average operations per second = 178

Insert 100 rows into key table (delete all rows before insert) gave :-

7.43 seconds real time

Total time = 7.43

Average operations per second = 13

Select 100 rows using primary key :-

0.63 seconds real time

Total time = .63

Average operations per second = 158

Insert 100 rows into new non-key table gave :-

0.47 seconds real time

Total time = .47

Average operations per second = 212

Insert 100 rows into non-key table (delete all rows before insert) gave :-

7.43 seconds real time

Total time = 7.43

Average operations per second = 13

Select 100 rows without a key :-

0.69 seconds real time

Total time = .69

Average operations per second = 144

2. ทดสอบเพิ่มข้อมูลและสอบถามข้อมูล 500 ครั้ง

Test SQL Prototype Performance. Test machine = FreeBSD phueng.ac.th 2.0.5-

RELEASE FreeBSD 2.0.5-RELEASE #0: Sat Jun 10 10:46:56 1995

jkh@westhill.cdrom.com:/usr/src/sys/compile/GENERIC i386

Insert 500 rows into new key table gave :-

2.56 seconds real time

Total time = 2.56

Average operations per second = 195

Insert 500 rows into key table (delete all rows before insert) gave :-

39.07 seconds real time

Total time = 39.07

Average operations per second = 12

Select 500 rows using primary key :-

3.25 seconds real time

Total time = 3.25

Average operations per second = 153

Insert 500 rows into new non-key table gave :-

1.99 seconds real time

Total time = 1.99

Average operations per second = 251

Insert 500 rows into non-key table (delete all rows before insert) gave :-

33.96 seconds real time

Total time = 33.96

Average operations per second = 14

Select 500 rows without a key :-

5.04 seconds real time

Total time = 5.04

Average operations per second = 99

3. ทดสอบเพิ่มข้อมูลและสอบถามข้อมูล 1000 ครั้ง

Test SQL Prototype Performance. Test machine = FreeBSD phueng.ac.th 2.0.5-

RELEASE FreeBSD 2.0.5-RELEASE #0: Sat Jun 10 10:46:56 1995

jkh@westhill.cdrom.com:/usr/src/sys/compile/GENERIC i386

Insert 1000 rows into new key table gave :-

5.47 seconds real time

Total time = 5.47

Average operations per second = 182

Insert 1000 rows into key table (delete all rows before insert) gave :-

72.73 seconds real time

Total time = 72.73

Average operations per second = 13

Select 1000 rows using primary key :-

7.02 seconds real time

Total time = 7.02

Average operations per second = 142

Insert 1000 rows into new non-key table gave :-

3.85 seconds real time

Total time = 3.85

Average operations per second = 259

Insert 1000 rows into non-key table (delete all rows before insert) gave :-

77.21 seconds real time

Total time = 77.21

Average operations per second = 12

Select 1000 rows without a key :-

13.72 seconds real time

Total time = 13.72

Average operations per second = 72

4. ทดสอบเพิ่มข้อมูลและสอบถามข้อมูล 2000 ครั้ง

Test SQL Prototype Performance. Test machine = FreeBSD phueng.ac.th 2.0.5-

RELEASE FreeBSD 2.0.5-RELEASE #0: Sat Jun 10 10:46:56 1995

jkh@westhill.cdrom.com:/usr/src/sys/compile/GENERIC i386

Insert 2000 rows into new key table gave :-

12.58 seconds real time

Total time = 12.58

Average operations per second = 158

Insert 2000 rows into key table (delete all rows before insert), gave :-

145.79 seconds real time

Total time = 145.79

Average operations per second = 13

Select 2000 rows using primary key :-

15.96 seconds real time

Total time = 15.96

Average operations per second = 125

Insert 2000 rows into new non-key table gave :-

7.67 seconds real time

Total time = 7.67

Average operations per second = 260

Insert 2000 rows into non-key table (delete all rows before insert) gave :-

156.17 seconds real time

Total time = 156.17

Average operations per second = 12

Select 2000 rows without a key :-

41.91 seconds real time

Total time = 41.91

Average operations per second = 47

4. ทดสอบเพิ่มข้อมูลและสอบถามข้อมูล 5000 ครั้ง

Test SQL Prototype Performance. Test machine = FreeBSD phueng.ac.th 2.0.5-

RELEASE FreeBSD 2.0.5-RELEASE #0: Sat Jun 10 10:46:56 1995

jkh@westhill.cdrom.com:/usr/src/sys/compile/GENERIC i386

Insert 5000 rows into new key table gave :-

45.23 seconds real time

Total time = 45.23

Average operations per second = 110

Insert 5000 rows into key table (delete all rows before insert) gave :-

390.92 seconds real time

Total time = 390.92

Average operations per second = 12

Select 5000 rows using primary key :-

51.75 seconds real time

Total time = 51.75

Average operations per second = 96

Insert 5000 rows into new non-key table gave :-

19.74 seconds real time

Total time = 19.74

Average operations per second = 253

Insert 5000 rows into non-key table (delete all rows before insert) gave :-

387.24 seconds real time

Total time = 387.24

Average operations per second = 12

Select 5000 rows without a key :-

217.24 seconds real time

Total time = 217.24

Average operations per second = 23

5. ทดสอบเพิ่มข้อมูลและสอบถามข้อมูล 7000 ครั้ง

Test SQL Prototype Performance. Test machine = FreeBSD phueng.ac.th 2.0.5-

RELEASE FreeBSD 2.0.5-RELEASE #0: Sat Jun 10 10:46:56 1995

jkh@westhill.cdrom.com:/usr/src/sys/compile/GENERIC i386

Insert 7000 rows into new key table gave :-

75.28 seconds real time

Total time = 75.28

Average operations per second = 92

Insert 7000 rows into key table (delete all rows before insert) gave :-

549.65 seconds real time

Total time = 549.65

Average operations per second = 12

Select 7000 rows using primary key :-

83.34 seconds real time

Total time = 83.34

Average operations per second = 83

Insert 7000 rows into new non-key table gave :-

27.70 seconds real time

Total time = 27.70

Average operations per second = 252

Insert 7000 rows into non-key table (delete all rows before insert) gave :-

524.85 seconds real time

Total time = 524.85

Average operations per second = 13

Select 7000 rows without a key :-

406.74 seconds real time

Total time = 406.74

Average operations per second = 17

6. ทดสอบเพิ่มข้อมูลและสอบถามข้อมูล 10000 ครั้ง

Test SQL Prototype Performance. Test machine = FreeBSD phueng.ac.th 2.0.5-

RELEASE FreeBSD 2.0.5-RELEASE #0: Sat Jun 10 10:46:56 1995

jkh@westhill.cdrom.com:/usr/src/sys/compile/GENERIC i386

Insert 10000 rows into new key table gave :-

133.64 seconds real time

Total time = 133.64

Average operations per second = 74

Insert 10000 rows into key table (delete all rows before insert) gave :-

772.66 seconds real time

Total time = 772.66

Average operations per second = 12

Select 10000 rows using primary key :-

142.29 seconds real time

Total time = 142.29

Average operations per second = 70

Insert 10000 rows into new non-key table gave :-

39.54 seconds real time

Total time = 39.54

Average operations per second = 252

Insert 10000 rows into non-key table (delete all rows before insert) gave :-

800.12 seconds real time

Total time = 800.12

Average operations per second = 12

Select 10000 rows without a key :-

811.66 seconds real time

Total time = 811.66

Average operations per second = 12

7. ทดสอบเพิ่มข้อมูลและสอบถามข้อมูล 50000 ครั้ง

Test SQL Prototype Performance. Test machine = FreeBSD phueng.ac.th 2.0.5-

RELEASE FreeBSD 2.0.5-RELEASE #0: Sat Jun 10 10:46:56 1995

jkh@westhill.cdrom.com:/usr/src/sys/compile/GENERIC i386

Insert 50000 rows into new key table gave :-

2442.18 seconds real time

Total time = 2442.18

Average operations per second = 20

Insert 50000 rows into key table (delete all rows before insert) gave :-

7700.80 seconds real time

Total time = 7700.80

Average operations per second = 6

Select 50000 rows using primary key :-

2311.18 seconds real time

Total time = 2311.18

Average operations per second = 21

Insert 50000 rows into new non-key table gave :-

213.89 seconds real time

Total time = 213.89

Average operations per second = 233

Insert 50000 rows into non-key table (delete all rows before insert) gave :-

5050.07 seconds real time

Total time = 5050.07

Average operations per second = 9

Select 50000 rows without a key :-

19438.96 seconds real time

Total time = 19438.96

Average operations per second = 2

ประวัติผู้เขียน

ร้อยเอกสัญญา วงษ์มาก สำเร็จการศึกษาระดับปริญญาตรี บริหารธุรกิจบัณฑิต สาขาการบัญชี คณะบริหารธุรกิจ มหาวิทยาลัยรามคำแหง และเข้าศึกษาต่อในหลักสูตร วิทยาศาสตร์มหาบัณฑิต ที่ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อ พ.ศ. 2535

