

รายการอ้างอิง

ภาษาไทย

1. ญาณี กาศัย. จัดการระบบฐานข้อมูลอย่างมืออาชีพ Oracle DBA. กรุงเทพฯ:ด้านสุทธากา
รพิมพ์, 2546.
2. เปรมฤดี กังวานวงศ์. การเปรียบเทียบสมรรถนะการทำงานระหว่างฐานข้อมูลเชิงสัมพันธ์เชิง
วัตถุและฐานข้อมูลเชิงสัมพันธ์. วิทยานิพนธ์ปริญญาโทมหาบัณฑิต ภาควิชาวิทยาศาสตร์
คอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย. จุฬาลงกรณ์มหาวิทยาลัย,
2544.

ภาษาอังกฤษ

3. Chris Foot, An Oracle Instructor's Guide to Oracle10g. Available from :
<http://www.dbszine.com/foot10.shtml>.
4. Guy Harrison, Oracle SQL high-performance tuning. Prentice Hall PTR, 1997.
5. Herman D. Hughes .Measuring Performance ,Available from :
<http://www.cse.msu.edu>.
6. Mainstay Partners. Dell Consolidates European Support System with Oracle
Enterprise Grid on Dell. Making technology investments count.
7. Miranda Nash, Oracle 10g : Infrastructure of Grid Computing. Available from :
[http://www.oracle.com/technology/tech/grid/collateral/GridTechWhitePaper_final.
pdf](http://www.oracle.com/technology/tech/grid/collateral/GridTechWhitePaper_final.pdf):/2003
8. Oracle Corporation, Measuring Performance with the Oracle Dynamic Monitoring
System (DMS). Available from : [http://download-
west.oracle.com/otn_hosted_doc/toplink/1013/b13593/optimiz015.htm](http://download-west.oracle.com/otn_hosted_doc/toplink/1013/b13593/optimiz015.htm), 2004.
9. Oracle Corporation, New Feature Guide. Available from :
http://otn.oracle.com/pls/db10g/portal.portal_demo3?selected=1, Oracle
Corporation, 2003.
10. Oracle Corporation, Oracle Architecture. Available from :
<http://www.siue.edu/~dbock/cm565/ch1-architecture.htm>, Oracle Corporation,
2001.

11. Oracle Corporation. Oracle9i Database Performance Tuning. Oracle Corporation, August 2002.
12. Oracle Corporation, Oracle9i Database Performance Tuning, Oracle Corporation, 2003.
13. Oracle Thailand. Oracle9i New Feature for Administrators. Oracle Thailand, July 2002.
14. Rubenstein, W B, Kubicar, M S and Cattell, R G G. Benchmarking Simple Database Operations. Sun Microsystems, Incorporated, Mountain View, California.
15. Vijay Kumar, Performance of Concurrency Control Mechanisms in Centralized Database Systems.(n.p): Prentice-Hall, 1996.

ภาคผนวก

ภาคผนวก ก

โปรแกรมและคำสั่งเซลล์สคริปต์ร่วมกับเอสคิวแอลพลัส

โปรแกรมและคำสั่งเซลล์สคริปต์ร่วมกับเอสคิวแอลพลัส ที่ใช้ในการในกรณีศึกษาทั้ง

15 กรณีมีรายละเอียดดังต่อไปนี้

- 1) สคริปต์ในการสร้างตารางจากการใช้คำสั่ง *Create as select table*
step1_create_tab1.sh
- 2) สคริปต์ในการเรียกดูข้อมูล step2_select1.sql
- 3) สคริปต์ในการสร้างอินเด็กซ์ step3_create_ind1.sh
- 4) สคริปต์ในการเรียกดูข้อมูลโดยการใช้การเชื่อมต่อตาราง step4_join1.sh
- 5) สคริปต์ในการเพิ่มข้อมูลในปริมาณมาก step5_insert_parallel.sh
- 6) สคริปต์ในการลบข้อมูลออกจากตาราง step6_delete_parallel.sh
- 7) สคริปต์ในการปรับปรุงอินเด็กซ์ step7_rebuild_index.sh
- 8) สคริปต์ในการเพิ่มข้อมูลในปริมาณมาก step8_insert_partition.sh
- 9) สคริปต์ในการจัดเก็บข้อมูลประจำวันด้วยโปรแกรม SQL Loader
step9_load_transaction.sh
- 10) สคริปต์ในการปรับปรุงพาดิชนอินเด็กซ์ step10_rebuild_index.sh
- 11) สคริปต์ในการใช้ฟังก์ชันสรุปยอด step11_summary.sh
- 12) สคริปต์ในการจัดเรียงข้อมูล โดยการใช้คำสั่ง *Group by* และ *Order by*
step12_sort1.sh
- 13) สคริปต์ในการลบข้อมูลออกจากตาราง step13_export.sh
- 14) สคริปต์ในการลบข้อมูลออกจากตาราง step14_import.sh
- 15) สคริปต์ในการลบข้อมูลออกจากตาราง step15_analyze.sh

ก.1 สคริปต์ในการสร้างตารางจากการใช้คำสั่ง *Create as select table*

step1_create_tab1.sh

```
DT=`date +%Y%m%d_%H:%M` ;export DT
sqlplus -s user/password@ora92 @step1_create_tab1.sql >
step1_CREATE_tab1_$.log &
```

คำสั่ง SQL ที่ใช้ในการสร้างตาราง step1_create_tab1.sql

```
WHENEVER SQLERROR EXIT 16
```

```
timing start
set echo off
set pause off
set head off
set verify off
```

```
SELECT '## start at : '||TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss') FROM dual;
```

```
CREATE TABLE call_detail_test tablespace call_detail as SELECT * FROM call_detail_temp
/
```

```
SELECT '## End at : '||TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss') FROM dual;
timing stop
EXIT;
```

ก.2 สคริปในการเรียกดูข้อมูล step2_select1.sql

```
DT=`date +%Y%m%d_%H:%M` ;export DT
sqlplus -s user/password@ora92 @step2_select1.sql > step2_select1_$$DT.log &
```

คำสั่ง SQL ที่ใช้ในการเรียกดูข้อมูล step2_select1.sql

```
WHENEVER SQLERROR EXIT 16
```

```
timing start
set echo off
set pause off
set head off
set verify off
```

```
SELECT '## start at : '||TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss') FROM dual;
```

```
SELECT * FROM call_detail_test
WHERE mobile_no = '088163335'
/
```

```
SELECT '## End at : '||TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss') FROM dual;
timing stop
EXIT;
```

ก.3 สคริปในการสร้างอินเด็กซ์ step3_create_ind1.sh

```
DT=`date +%Y%m%d_%H:%M` ;export DT
sqlplus -s user/password@ora92 @step3_create_ind1.sql >
step3_create_ind1_$$DT.log &
```

คำสั่ง SQL ที่ใช้ในการสร้างอินเด็กซ์ step3_create_ind1.sql

```
WHENEVER SQLERROR EXIT 16
```

```
timing start
set echo off
set pause off
set head off
set verify off
```

```
SELECT '## start at : '||TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss') FROM dual;
```

```
CREATE INDEX paricm11.call_detail_test_idx
ON paricm11.call_detail_test (mobile_no)
tablespace call_detail_ind
/
```

```
SELECT '## End at : '||TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss') FROM dual;
timing stop
EXIT;
```

ก.4 สคริปในการเรียกดูข้อมูลโดยการใช้การเชื่อมต่อตาราง step4_join1.sh

```
DT=`date +%Y%m%d_%H:%M`;export DT
sqlplus -s user/password@ora92 @step4_join1.sql > step4_join1_&DT.log &
```

คำสั่ง SQL ที่ใช้ในการเรียกดูข้อมูลโดยการใช้การเชื่อมต่อตาราง step4_join1.sql

```
WHENEVER SQLERROR EXIT 16
```

```
timing start
set echo off
set pause off
set head off
set verify off
```

```
SELECT '## start at : '||TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss') FROM dual;
```

```
SELECT count(*)
--call.call_start_time,call.mobile_no,fea.feature_code,fea.description_engl
FROM call_detail_test call, feature fea
WHERE call.feature_code = fea.feature_code;
/
```

```
SELECT '## End at : '||TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss') FROM dual;
timing stop
EXIT;
```

ก.5 สคริปในการเพิ่มข้อมูลในปริมาณมาก `step5_insert_parallel.sh`

```
DT=`date +%Y%m%d_%H:%M` ;export DT
sqlplus -s user/password@ora92 @step5_insert_parallel.sql >
step5_insert_parallel_$$DT.log &
```

คำสั่ง SQL ที่ใช้ในการเพิ่มข้อมูลในปริมาณมาก `step5_insert_parallel.sql`

```
WHENEVER SQLERROR EXIT 16
```

```
timing start
set echo off
set pause off
set head off
set verify off
```

```
SELECT '## start at : '||TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss') FROM dual;
```

```
INSERT /*+ APPEND PARALLEL ( A, 2 ) */
  INTO call_detail_test a
  SELECT /*+ parallel (t,2) */ *
  FROM call_detail_20050815 t
```

```
/
COMMIT;
/
```

```
SELECT '## End at : '||TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss') FROM dual;
timing stop
EXIT;
```

ก.6 สคริปในการลบข้อมูลออกจากตาราง `step6_delete_parallel.sh`

```
DT=`date +%Y%m%d_%H:%M` ;export DT
sqlplus -s user/password@ora92 @step6_delete_parallel.sql >
step6_delete_parallel_$$DT.log &
```

คำสั่ง SQL ที่ใช้ในการลบข้อมูลออกจากตาราง `step6_delete_parallel.sql`

```
WHENEVER SQLERROR EXIT 16
```

```
timing start
set echo off
set pause off
set head off
set verify off
```

```
SELECT '## start at : '||TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss') FROM dual;
```

```
DELETE /*+ parallel(a,2) */ call_detail_test
WHERE trunc(call_start_time) = '15-aug-2005'
```

```
/
COMMIT;
/
```

```
SELECT '## End at : '||TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss') FROM dual;
timing stop
EXIT;
```

ก.7 สคริปในการปรับปรุงอินเด็กซ์ step7_rebuild_index.sh

```
DT=`date +%Y%m%d_%H:%M` ;export DT
sqlplus -s user/password@ora92 @step7_rebuild_index.sql >
step7_rebuild_index_${DT}.log &
```

คำสั่ง SQL ที่ใช้ในการปรับปรุงอินเด็กซ์ step7_rebuild_index.sql

```
WHENEVER SQLERROR EXIT 16
```

```
timing start
set echo off
set pause off
set head off
set verify off
```

```
SELECT '## start at : '||TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss') FROM dual;
```

```
ALTER INDEX PARICM11.CALL_DETAIL_TEST_IDX
REBUILD
NOCOMPRESS
NOLOGGING
TABLESPACE CALL_DETAIL_IND
/
```

```
SELECT '## End at : '||TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss') FROM dual;
timing stop
EXIT;
```

ก.8 สคริปในการเพิ่มข้อมูลในปริมาณมาก step8_insert_partition.sh

```
DT=`date +%Y%m%d_%H:%M` ;export DT
sqlplus -s user/password@ora92 @step8_insert_partition.sql >
step8_insert_partition_${DT}.log &
```

คำสั่ง SQL ที่ใช้ในการเพิ่มข้อมูลในปริมาณมาก step8_insert_partition.sql

```
WHENEVER SQLERROR EXIT 16
```

```
timing start
set echo off
set pause off
set head off
set verify off
```

```
ALTER SESSION SET skip_unusable_indexes = TRUE;
/
```



```
SELECT '## start at : '||TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss') FROM dual;
```

```
INSERT INTO CALL_DETAIL PARTITION(CALL_DETAIL_20050816) a
```

```
( CALL_START_TIME
,MOBILE_NO
,NETWORK_TYPE
,HOUR
,CALL_TYPE
,DAY_CODE
,TIME_BAND
,CALLED_B_NO
,NETWORK_TYPE_B_NO
,DISTANCE_BAND
,CALL_END_TIME
,CALL_DURATION
,UNITS
,FEATURE_CODE
,TERMINATE_REASON
,ZONE_A
,ZONE_B
,PP_CHARGE_AMOUNT
,BALANCE_AMOUNT
,PROCESS_DATE
,GROSS_AMOUNT
,FILE_DATE
,SERVICE_TYPE
,RATE
,FILE_SEQ
)
```

```
SELECT
CALL_START_TIME
,MOBILE_NO
,NETWORK_TYPE
,HOUR
,CALL_TYPE
,DAY_CODE
,TIME_BAND
,CALLED_B_NO
,NETWORK_TYPE_B_NO
,DISTANCE_BAND
,CALL_END_TIME
,CALL_DURATION
,UNITS
,FEATURE_CODE
,TERMINATE_REASON
,ZONE_A
,ZONE_B
,PP_CHARGE_AMOUNT
,BALANCE_AMOUNT
,PROCESS_DATE
,GROSS_AMOUNT
,TO_DATE('20050816','YYYYMMDD') FILE_DATE
,SERVICE_TYPE
,RATE
```

```
,FILE_SEQ
FROM CALL_DETAIL_TEST H
/
COMMIT;
```

```
SELECT '## End at : '|TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss') FROM dual;
timing stop
EXIT;
```

ก.9 สคริปในการจัดเก็บข้อมูลประจำวันด้วยโปรแกรม SQL Loader

step9 load transaction.sh

```
echo '#####'
echo '## Module: Load transaction into table call_detail ##'
echo '#####'
echo '## Step1 : Copy file to /tmp and uncompress file ##'
echo '## Step2 : Load cdr (cdr_yyyymmdd.txt) to temp table CALL_DETAIL_TEST##'
echo '## Step3 : Insert data into table CALL_DETAIL ##'
echo '#####'
echo ''
DT=`date +%d%m%Y_%H:%M`
echo "## START COPY1 AT: $DT "
echo '#####'
echo '## STEP1: Copy file /tmp and uncompress file'
echo '#####'
echo 'Copy file /export/home/paricm11/data/cdr_20050815_1.Z to /tmp'
echo 'Working.....'
cp /export/home/paricm11/data/cdr_20050815_1.Z /tmp/data/cdr_20050815.Z
cd /tmp/data/
uncompress cdr_20050815.Z
mv cdr_20050815 cdr_20050815.txt
DT=`date +%d%m%Y_%H:%M`
echo "## END COPY1 AT : $DT"
echo '#####'
echo '## STEP2 : Load cdr file to table CALL_DETAIL #'
echo '#####'
DT=`date +%d%m%Y_%H:%M`
echo "## START LOAD DATA1 AT: $DT "
sqlldr parfile=/export/home/paricm11/data/call_detail.par
silent=feedback,discards,header
DT=`date +%d%m%Y_%H:%M`
echo "## END LOAD DATA1 AT : $DT"
rtn=$?
if [ $rtn -ne 0 ]; then
    echo ' >>> ERROR : Run step Load cdr file number 1 '
    exit 1
fi
rm /tmp/data/cdr*
DT=`date +%d%m%Y_%H:%M`
echo ''
echo "## START COPY2 AT: $DT "
```

```

echo '#####'
echo '## STEP1: Copy file /tmp and uncompress file'
echo '#####'
echo 'Copy file /export/home/paricm11/data/cdr_20050815_2.Z to /tmp'
echo 'Working.....'
cp /export/home/paricm11/data/cdr_20050815_2.Z /tmp/data/cdr_20050815.Z
cd /tmp/data/
uncompress cdr_20050815.Z
mv cdr_20050815 cdr_20050815.txt
DT=`date +%d%m%Y_%H:%M`
echo "## END COPY2 AT : $DT"
echo '#####'
echo '## STEP2 : Load cdr file to table CALL_DETAIL #'
echo '#####'
DT=`date +%d%m%Y_%H:%M`
echo "## START LOAD DATA2 AT: $DT "
sqlldr parfile=/export/home/paricm11/data/call_detail.par
silent=feedback,discards,header
DT=`date +%d%m%Y_%H:%M`
echo "## END LOAD DATA2 AT : $DT"
rtn=$?
if [ $rtn -ne 0 ]; then
    echo ' >>> ERROR : Run step Load cdr file number 2 '
    exit 1
fi
rm /tmp/data/cdr*
DT=`date +%d%m%Y_%H:%M`
echo ''
echo "## START COPY3 AT: $DT "
echo '#####'
echo '## STEP1: Copy file /tmp and uncompress file'
echo '#####'
echo 'Copy file /export/home/paricm11/data/cdr_20050815_3.Z to /tmp'
echo 'Working.....'
cp /export/home/paricm11/data/cdr_20050815_3.Z /tmp/data/cdr_20050815.Z
cd /tmp/data/
uncompress cdr_20050815.Z
mv cdr_20050815 cdr_20050815.txt
DT=`date +%d%m%Y_%H:%M`
echo "## END COPY2 AT : $DT"
echo '#####'
echo '## STEP2 : Load cdr file to table CALL_DETAIL #'
echo '#####'
DT=`date +%d%m%Y_%H:%M`
echo "## START LOAD DATA3 AT: $DT "
sqlldr parfile=/export/home/paricm11/data/call_detail.par
silent=feedback,discards,header
DT=`date +%d%m%Y_%H:%M`
echo "## END LOAD DATA3 AT : $DT"
rtn=$?
if [ $rtn -ne 0 ]; then
    echo ' >>> ERROR : Run step Load cdr file number 3 '
    exit 1
fi

```

```

rm /tmp/data/cdr*
DT=`date +%d%m%Y_%H:%M`
echo ''
echo "## START COPY4 AT: $DT "
echo '#####'
echo '## STEP1: Copy file /tmp and uncompress file'
echo '#####'
echo 'Copy file /export/home/paricm11/data/cdr_20050815_4.Z to /tmp'
echo 'Working.....'
cp /export/home/paricm11/data/cdr_20050815_4.Z /tmp/data/cdr_20050815.Z
cd /tmp/data/
uncompress cdr_20050815.Z
mv cdr_20050815 cdr_20050815.txt
DT=`date +%d%m%Y_%H:%M`
echo "## END COPY4 AT : $DT"
echo '#####'
echo '## STEP2 : Load cdr file to table CALL_DETAIL #'
echo '#####'
DT=`date +%d%m%Y_%H:%M`
echo "## START LOAD DATA4 AT: $DT "
sqlldr parfile=/export/home/paricm11/data/call_detail.par
silent=feedback,discards,header
DT=`date +%d%m%Y_%H:%M`
echo "## END LOAD DATA4 AT : $DT"
rtn=$?
if [ $rtn -ne 0 ]; then
    echo ' >>> ERROR : Run step Load cdr file number 4 '
    exit 1
fi
rm /tmp/data/cdr*
DT=`date +%d%m%Y_%H:%M`
echo ''
echo '#####'
echo '## STEP3 : Insert data into table CALL_DETAIL #'
echo '#####'
sqlplus -s paricm11/ducan2515@ora92
@/export/home/paricm11/data/script_insert_data.sql
exit 0

```

ก.10 สคริปในการปรับปรุงพาดิชั้นอินเด็กซ์ step10_rebuild_index.sh

```

DT=`date +%Y%m%d_%H:%M` ;export DT
sqlplus -s user/password@ora92 @step10_rebuild_index.sql >
step10_rebuild_index_`$DT`.log &

```

คำสั่ง SQL ที่ใช้ในการปรับปรุงพาดิชนีในเด็กซ์ step10_rebuild_index.sql

```
WHENEVER SQLERROR EXIT 16
```

```
timing start
set echo off
set pause off
set head off
set verify off
```

```
SELECT '## start at : '||TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss') FROM dual;
```

```
ALTER INDEX call_detail_ix rebuild partition call_detail_ind_20050816
NOCOMPRESS
NOLOGGING
TABLESPACE CALL_DETAIL_IND
/
```

```
SELECT '## End at : '||TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss') FROM dual;
timing stop
EXIT;
```

ก.11 สคริปในการใช้ฟังก์ชันสรุปยอด step11_summary.sh

```
DT=`date +%Y%m%d_%H:%M`;export DT
sqlplus -s user/password@ora92 @step11_summary.sql > step11_summary_&DT.log &
```

คำสั่ง SQL ที่ใช้ในการใช้ฟังก์ชันสรุปยอด step11_summary.sql

```
WHENEVER SQLERROR EXIT 16
DROP TABLE sum_call_20050815
/
```

```
SELECT '## Start time CREATE TABLE : '||TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss')
FROM dual;
timing start
```

```
CREATE TABLE SUM_CALL_&&1
(
DAY_CODE DATE
,MOBILE_NO VARCHAR2(10)
,CUSTOMER_ID NUMBER(10)
,REGION_ORIGINATE VARCHAR2(2)
,REGION_DISTINATE VARCHAR2(2)
,HOUR NUMBER
,DAY_TYPE VARCHAR2(4)
,NETWORK_TYPE_B_NO VARCHAR2(3)
,DISTANCE_BAND VARCHAR2(1)
,FEATURE_CODE NUMBER
,CALLED_B_NO VARCHAR2(10)
,SERVICE_TYPE VARCHAR2(5)
,TOTAL_CALL NUMBER
,TOTAL_MINUTE NUMBER
```

```

,TOTAL_SECOND          NUMBER
,TOTAL_AMOUNT          NUMBER
,FREEDOM_BONUS         VARCHAR2(1)
,CLASS_OF_SERVICE      NUMBER(3)
)
TABLESPACE CALL_DETAIL;

timing start
INSERT INTO SUM_CALL_&&1
( day_code, mobile_no, customer_id, region_originate, region_distinate, hour,
  day_type, network_type_b_no, distance_band, feature_code,
  total_call, total_minute, total_second, total_amount
)
SELECT
day_code, mobile_no, customer_id, region_originate, region_distinate, hour,
day_type, network_type_b_no, distance_band, feature_code,
sum(call) total_call, sum(min) total_minute, sum(sec) total_second, sum(amount)
total_amount
FROM
(
SELECT
  TO_DATE('&&1','YYYYMMDD') day_code,
  mobile_no,
  customer_id,
  DECODE(zone_a,1,'CB',2,'XU',3,'XL',4,'NU',5,'NL',6,'SU',7,'SL','XX') region_originate,
  DECODE(substr(zone_b,-1,1),1,'CB',2,'XU',3,'XL',4,'NU',5,'NL',6,'SU',7,'SL','XX')
    region_distinate,
  nvl(hour,99) hour,
  DECODE(TO_CHAR(call_start_time,'DY'),'SUN','SUN','SAT','SAT','WEEK') day_type,
  nvl(network_type_b_no,'XXX') network_type_b_no,
  DECODE(distance_band,'1','1','2','2','3','3','4','4','0') distance_band,
  DECODE(feature_code,137,29,30,29,54,45, 55,45, 56,45, 28,22,79,25,feature_code)
    feature_code,
  DECODE(zone_a,99,DECODE(service_type,'1',0,1),
    DECODE(sign(pp_charge_amount),-1,-1,1)) call,
  DECODE(zone_a,99,DECODE(service_type,'1',
    DECODE(pp_charge_amount,0,0,units),units),units) min,
  DECODE(zone_a,99,DECODE(service_type,'1',
    DECODE(pp_charge_amount,0,0,call_duration),call_duration),
    call_duration) sec,
  pp_charge_amount/100 amount
FROM CALL_DETAIL partition(CALL_DETAIL_&&1)
)
GROUP BY day_code, mobile_no, customer_id,region_originate, region_distinate, hour,
  day_type, network_type_b_no, distance_band, feature_code
/

SELECT '## End time CREATE TABLE: '||TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss') FROM
dual;
timing stop
EXIT;

```

ก.12 สคริปในการจัดเรียงข้อมูล โดยการใช้คำสั่ง *Group by* และ *Order by*

step12_sort1.sh

```
DT=`date +%Y%m%d_%H:%M` ;export DT
sqlplus -s user/password@ora92 @step12_sort1.sql > step12_sort1_$$DT.log &
```

คำสั่ง SQL ที่ใช้ในการจัดเรียงข้อมูล โดยการใช้คำสั่ง *Group by* และ *Order by*

```
WHENEVER SQLERROR EXIT 16
```

```
SELECT '## Start time CREATE TABLE : '|TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss')
FROM dual;
timing start
```

```
SELECT network_type_b_no network_destination,
       count(*) transaction,
       sum(pp_charge_amount) sum_amount
FROM call_detail
WHERE trunc(call_start_time) = '15-aug-2005'
GROUP BY network_type_b_no
ORDER BY network_type_b_no
/
```

```
SELECT '## End time CREATE TABLE: '|TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss') FROM
dual;
timing stop
EXIT;
```

ก.13 สคริปในการลบข้อมูลออกจากตาราง step13_export.sh

```
DT=`date +%Y%m%d_%H:%M` ;export DT
sqlplus -s user/password@ora92 @step13_export.sql > step13_export_$$DT.log &
```

คำสั่ง SQL ที่ใช้ในการลบข้อมูลออกจากตาราง step13_export.sql

```
DT=`date +%Y%m%d_%H:%M:%S` ;export DT
echo 'Date start' $$DT
exp userid=username/password@ora92 tables=paricm11.call_detail:call_detail_20050815
file=/tmp/data/call_detail.dmp direct=y indexes=n log=call_detail_dmp_$$DT.log
DT=`date +%Y%m%d_%H:%M:%S` ;export DT
echo 'Date Stop' $$DT
```

ก.14 สคริปในการลบข้อมูลออกจากตาราง step14_import.sh

```
DT=`date +%Y%m%d_%H:%M` ;export DT
sqlplus -s user/password@ora92 @step14_import.sql > step14_import_$DT.log &
```

คำสั่ง SQL ที่ใช้ในการลบข้อมูลออกจากตาราง step14_import.sql

```
DT=`date +%Y%m%d_%H:%M:%S` ;export DT
echo 'Date start' $DT
imp userid=paricm11/duan2515@ora92 FROMuser=paricm11 touser=paricm11
tables=call_detail:call_detail_20050815 file=/tmp/data/call_detail.dmp indexes=n
constraints=n log=imp_call_detail_$DT.log ignore=y
DT=`date +%Y%m%d_%H:%M:%S` ;export DT
echo 'Date stop' $DT
```

ก.15 สคริปในการลบข้อมูลออกจากตาราง step15_analyze.sh

```
DT=`date +%Y%m%d_%H:%M` ;export DT
sqlplus -s user/password@ora92 @step15_analyze.sql > step15_analyze_$DT.log &
```

คำสั่ง SQL ที่ใช้ในการลบข้อมูลออกจากตาราง step15_analyze.sql

```
WHenever SQLERROR EXIT 16
```

```
timing start
set echo off
set pause off
set head off
set verify off
```

```
SELECT '## start at : '||TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss') FROM dual;
```

```
ANALYZE TABLE call_detail COMPUTE STATISTICS
/
```

```
SELECT '## End at : '||TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss') FROM dual;
timing stop
EXIT;
```


ภาคผนวก ข

โปรแกรมและคำสั่งเชลล์สคริปต์ร่วมกับเอสคิวแอลพลัสแบบเพิ่มประสิทธิภาพ

โปรแกรมและคำสั่งเชลล์สคริปต์ร่วมกับเอสคิวแอลพลัสแบบเพิ่มประสิทธิภาพตาม
กรณีศึกษาทั้ง 6 กรณีมีรายละเอียด ดังต่อไปนี้

- 1) สคริปต์ในการเรียกดูข้อมูล step2_select2.sql
- 2) สคริปต์ในการสร้างอินเด็กซ์ step3_create_ind2.sh
- 3) สคริปต์ในการปรับปรุงอินเด็กซ์ step7_rebuild_index2.sh
- 4) สคริปต์ในการปรับปรุงพาทิชันอินเด็กซ์ step10_rebuild_index2.sh
- 5) สคริปต์ในการใช้ฟังก์ชันสรุปยอด step11_summary_tune.sh
- 6) สคริปต์ในการจัดเรียงข้อมูล โดยการใช้คำสั่ง *Group by* และ *Order by*
step12_sort1_tune.sh

ข.1 สคริปต์ในการเรียกดูข้อมูล step2_select2.sql

```
DT=`date +%Y%m%d_%H:%M` ;export DT  
sqlplus -s user/password@ora92 @step2_select2.sql > step2_select2_$.log &
```

คำสั่ง SQL ที่ใช้ในการเรียกดูข้อมูล step2_select2.sql

```
WHenever SQLERROR EXIT 16
```

```
timing start  
set echo off  
set pause off  
set head off  
set verify off
```

```
SELECT '## start at : '||TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss') FROM dual;
```

```
SELECT /*+ parallel(a,4) */ * FROM call_detail_test a  
WHERE mobile_no = '088163335'  
/
```

```
SELECT '## End at : '||TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss') FROM dual;  
timing stop  
EXIT;
```

ข.2 สคริปในการสร้างอินเด็กซ์ step3_create_ind2.sh

```
DT=`date +%Y%m%d_%H:%M` ;export DT
sqlplus -s user/password@ora92 @step3_create_ind2.sql >
step3_create_ind2_$.log &
```

คำสั่ง SQL ที่ใช้ในการสร้างอินเด็กซ์ step3_create_ind2.sql

```
WHENEVER SQLERROR EXIT 16
```

```
timing start
set echo off
set pause off
set head off
set verify off
```

```
SELECT '## start at : '||TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss') FROM dual;
```

```
CREATE INDEX paricm11.call_detail_test_idx
ON paricm11.call_detail_test (mobile_no)
TABLESPACE call_detail_ind parallel 2
/
```

```
SELECT '## End at : '||TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss') FROM dual;
timing stop
EXIT;
```

ข.3 สคริปในการปรับปรุงอินเด็กซ์ step7_rebuild_index2.sh

```
DT=`date +%Y%m%d_%H:%M` ;export DT
sqlplus -s user/password@ora92 @step7_rebuild_index2.sql >
step7_rebuild_index2_$.log &
```

คำสั่ง SQL ที่ใช้ในการปรับปรุงอินเด็กซ์ step7_rebuild_index2.sql

```
WHENEVER SQLERROR EXIT 16
```

```
timing start
set echo off
set pause off
set head off
set verify off
```

```
SELECT '## start at : '||TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss') FROM dual;
```

```
ALTER INDEX PARICM11.CALL_DETAIL_TEST_IDX
REBUILD
NOCOMPRESS
PARALLEL 2
NOLOGGING
TABLESPACE CALL_DETAIL_IND
/
```

```
SELECT '## End at : '||TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss') FROM dual;
timing stop
EXIT;
```

ข.4 สคริปในการปรับปรุงพาดิชั่นอินเด็กซ์ step10_rebuild_index2.sh

```
DT=`date +%Y%m%d_%H:%M`;export DT
sqlplus -s user/password@ora92 @step10_rebuild_index2.sql >
step10_rebuild_index2_&DT.log &
```

คำสั่ง SQL ที่ใช้ในการปรับปรุงพาดิชั่นอินเด็กซ์ step10_rebuild_index2.sql

```
WHENEVER SQLERROR EXIT 16
```

```
timing start
set echo off
set pause off
set head off
set verify off
```

```
SELECT '## start at : '||TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss') FROM dual;
```

```
ALTER INDEX call_detail_ix rebuild partition call_detail_ind_20050816
NOCOMPRESS
PARALLEL 2
NOLOGGING
TABLESPACE CALL_DETAIL_IND
/
```

```
SELECT '## End at : '||TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss') FROM dual;
timing stop
EXIT;
```

ข.5 สคริปในการใช้ฟังก์ชันสรุปยอด step11_summary_tune.sh

```
DT=`date +%Y%m%d_%H:%M`;export DT
sqlplus -s user/password@ora92 @step11_summary_tune.sql >
step11_summary_tune_&DT.log &
```

คำสั่ง SQL ที่ใช้ในการใช้ฟังก์ชันสรุปยอด step11_summary_tune.sql

```
WHENEVER SQLERROR EXIT 16
DROP table SUM_call_20050815
/
```

```
SELECT '## Start time create table : '||TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss') FROM
dual;
timing start
```



```

CREATE TABLE SUM_CALL_&&1
(
  DAY_CODE          DATE
,MOBILE_NO         VARCHAR2(10)
,CUSTOMER_ID       NUMBER(10)
,REGION_ORIGINATE  VARCHAR2(2)
,REGION_DISTINATE  VARCHAR2(2)
,HOUR              NUMBER
,DAY_TYPE          VARCHAR2(4)
,NETWORK_TYPE_B_NO VARCHAR2(3)
,DISTANCE_BAND     VARCHAR2(1)
,FEATURE_CODE      NUMBER
,CALLED_B_NO       VARCHAR2(10)
,SERVICE_TYPE      VARCHAR2(5)
,TOTAL_CALL        NUMBER
,TOTAL_MINUTE      NUMBER
,TOTAL_SECOND      NUMBER
,TOTAL_AMOUNT      NUMBER
,FREEDOM_BONUS     VARCHAR2(1)
,CLASS_OF_SERVICE  NUMBER(3)
)
TABLESPACE CALL_DETAIL;

timing start
INSERT /*+ APPEND */
INTO SUM_CALL_&&1
( day_code, mobile_no, customer_id, region_originate, region_distinate, hour,
  day_type, network_type_b_no, distance_band, feature_code,
  total_call, total_minute, total_second, total_amount
)
SELECT /*+ PARALLEL(TMP,2) */
day_code, mobile_no, customer_id, region_originate, region_distinate, hour,
day_type, network_type_b_no, distance_band, feature_code,
SUM(call) total_call, SUM(min) total_minute, SUM(sec) total_second, SUM(amount)
total_amount
FROM
(
  SELECT /*+ FULL(PP) PARALLEL(PP,2) */
  TO_DATE('&&1','YYYYMMDD') day_code,
  mobile_no,
  customer_id,
  DECODE(zone_a,1,'CB',2,'XU',3,'XL',4,'NU',5,'NL',6,'SU',7,'SL','XX') region_originate,
  DECODE(substr(zone_b,-1,1),1,'CB',2,'XU',3,'XL',4,'NU',5,'NL',6,'SU',7,'SL','XX')
  region_distinate,
  nvl(hour,99) hour,
  DECODE(TO_CHAR(call_start_time,'DY'),'SUN','SUN','SAT','SAT','WEEK') day_type,
  nvl(network_type_b_no,'XXX') network_type_b_no,
  DECODE(distance_band, '1','1', '2','2', '3','3', '4','4', '0') distance_band,
  DECODE(feature_code, 137,29,30,29, 54,45, 55,45, 56,45, 28,22, 79,25, feature_code)
  feature_code,
  DECODE(zone_a,99,DECODE(service_type,'1',0,1),
  DECODE(sign(pp_charge_amount),-1,-1,1)) call,
  DECODE(zone_a,99,DECODE(service_type,'1',
  DECODE(pp_charge_amount,0,0,units),units),units) min,

```

```

        DECODE(zone_a,99,DECODE(service_type,'1',
            DECODE(pp_charge_amount,0,0,call_duration),call_duration),
            call_duration) sec,
        pp_charge_amount/100 amount
    FROM CALL_DETAIL partition(CALL_DETAIL_&&1) PP )
    GROUP BY day_code, mobile_no, customer_id,region_originate, region_distinate, hour,
        day_type, network_type_b_no, distance_band, feature_code
    /

```

```

SELECT '## End time create table: '||TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss') FROM
dual;
timing stop
EXIT;

```

ข.6 สคริปในการจัดเรียงข้อมูล โดยการใช้คำสั่ง *Group by* และ *Order by*

step12 sort1 tune.sh

```

DT=`date +%Y%m%d_%H:%M`;export DT
sqlplus -s user/password@ora92 @step12 sort1 tune.sql > step12_sort1_tune
_$.log &

```

คำสั่ง SQL ที่ใช้ในการจัดเรียงข้อมูล โดยการใช้คำสั่ง *Group by* และ *Order by*

```

WHENEVER SQLERROR EXIT 16

```

```

SELECT '## Start time CREATE TABLE : '||TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss')
FROM dual;
timing start

```

```

SELECT /*+ parallel(a,2) */
        network_type_b_no network_destination,
        count(*) transaction,
        sum(pp_charge_amount) sum_amount
    FROM call_detail detail partition(call_detail_20050815) a
    GROUP BY network_type_b_no
    ORDER BY network_type_b_no
    /

```

```

SELECT '## End time CREATE TABLE: '||TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss') FROM
dual;
timing stop
EXIT;

```

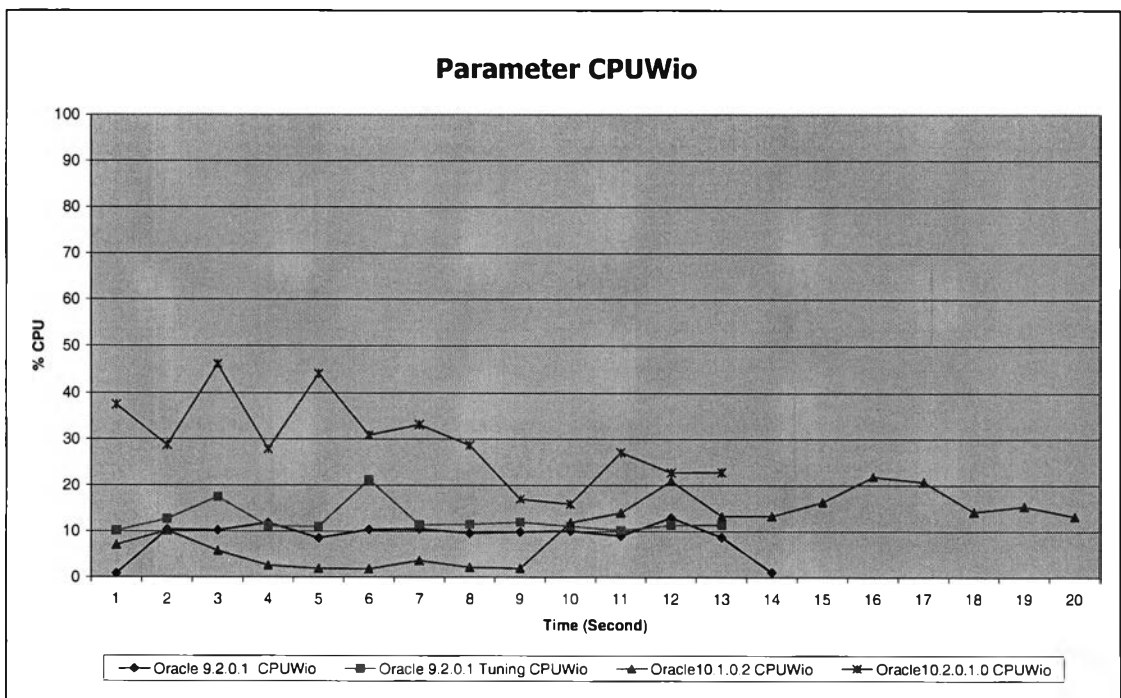
ภาคผนวก ค

ผลการวิจัยในด้านของการใช้ทรัพยากร

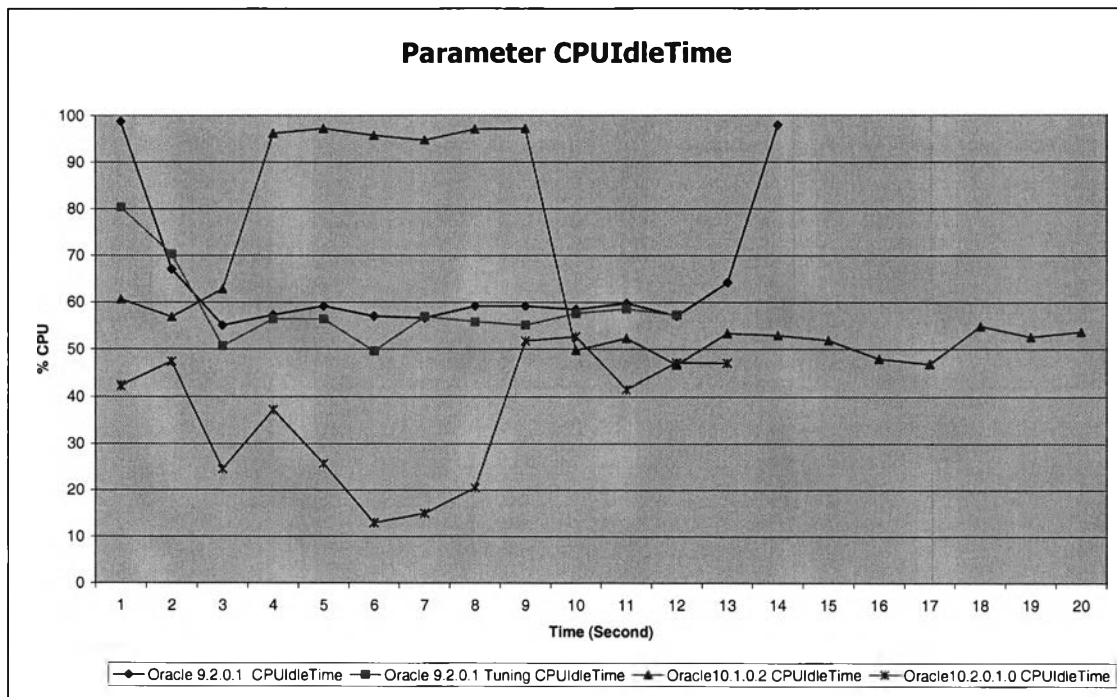
ผลของการใช้ทรัพยากรร่วมกันระหว่าง การใช้งานของซีพียู การใช้งานของจานแม่เหล็ก และการใช้งานของหน่วยความจำ

การแสดงผลนี้เป็นการแสดงผลของการใช้ทรัพยากรร่วมกันระหว่าง การใช้งานของซีพียู การใช้งานของจานแม่เหล็ก และการใช้งานของหน่วยความจำ โดยแสดงการเปรียบเทียบสมรรถนะของระบบจัดการฐานข้อมูลทั้ง 4 เวอร์ชันการทำงานที่ได้กล่าวข้างต้น แยกตามกรณีศึกษาดังต่อไปนี้

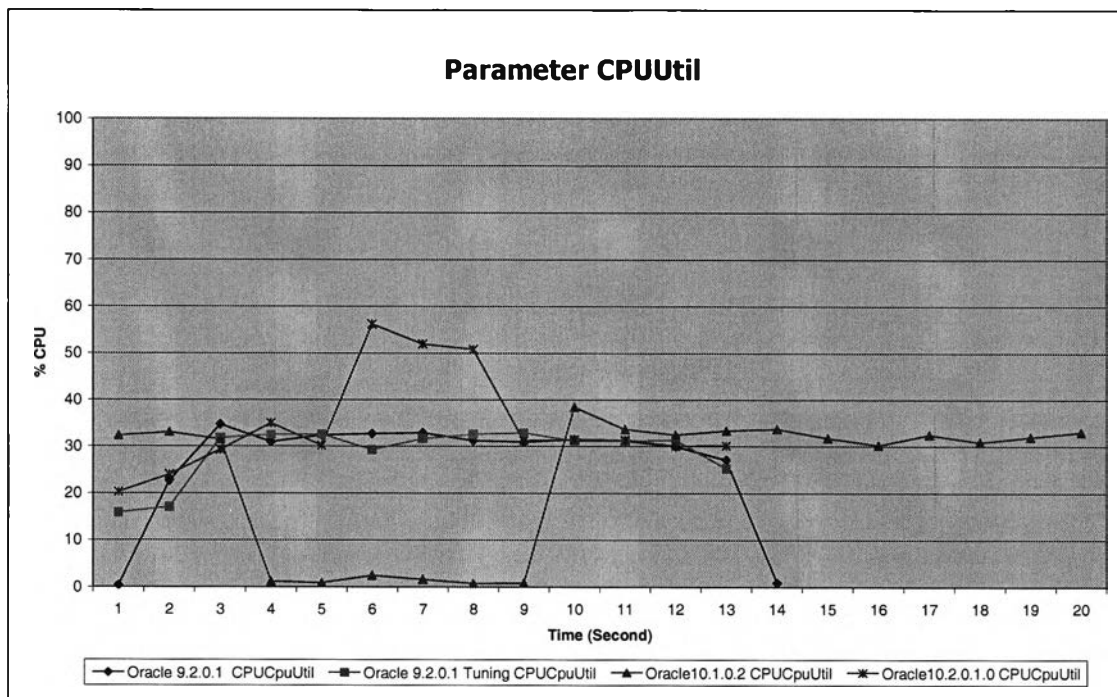
1) แสดงการสร้างตารางจากการใช้คำสั่ง *Create as select table* จะเห็นว่าระบบจัดการฐานข้อมูลออราเคิลเวอร์ชัน 10.2.0.1 มีการบริหารการใช้ทรัพยากรในส่วนซีพียูได้ดีกว่าเวอร์ชันอื่น และในส่วนของการใช้งานของดิสก์และการใช้หน่วยความจำก็มีปริมาณมากกว่าเวอร์ชันอื่น เช่นกัน โดยมีออราเคิลเวอร์ชัน 9.2.0.1 ที่มีการเพิ่มประสิทธิภาพการทำงานนั้นมีการบริหารการใช้ทรัพยากรรองลงมา



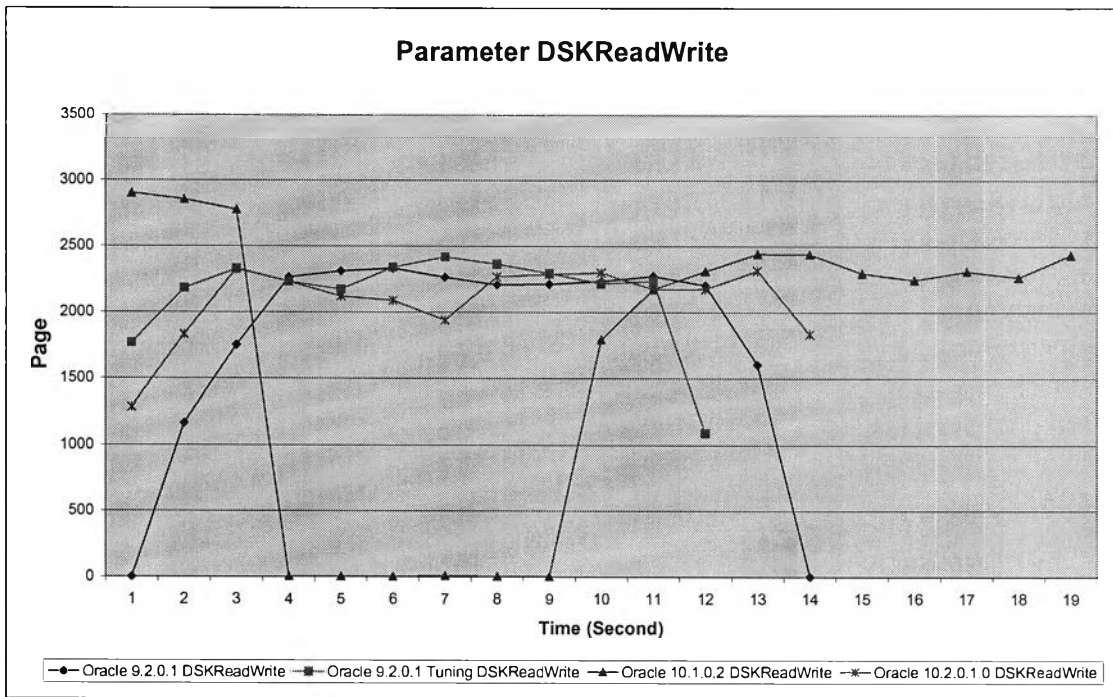
รูปที่ ค.1 แสดงการเปรียบเทียบพารามิเตอร์ CPUWio ในการสร้างตาราง



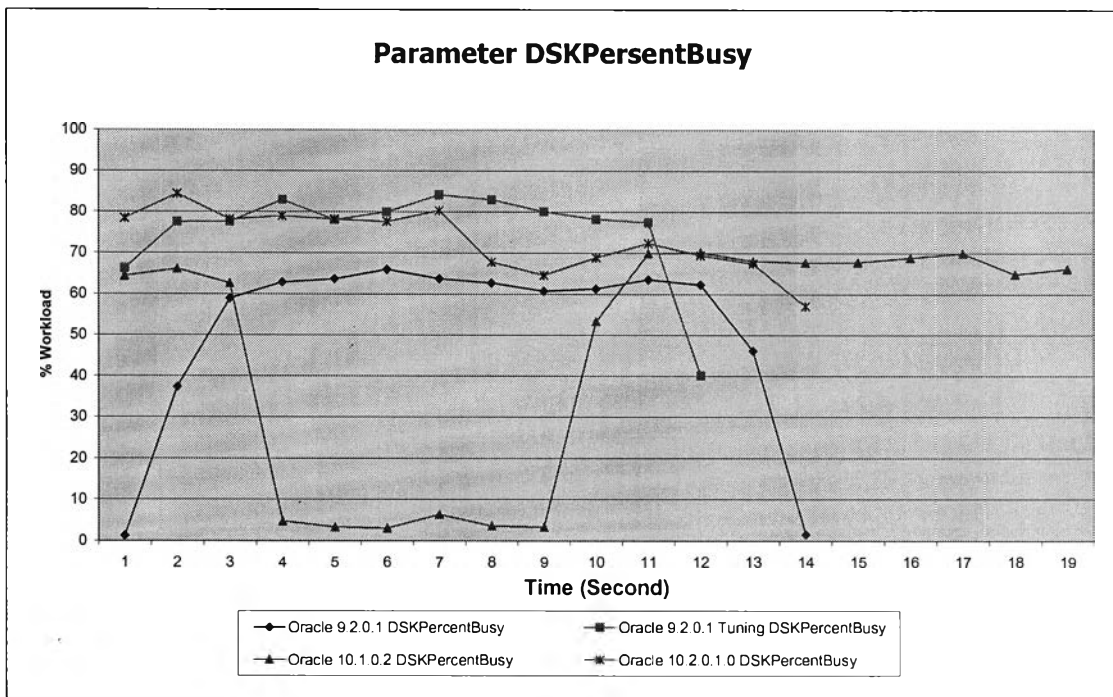
รูปที่ ค.2 แสดงการเปรียบเทียบพารามิเตอร์ CPUIdleTime ในการสร้างตาราง



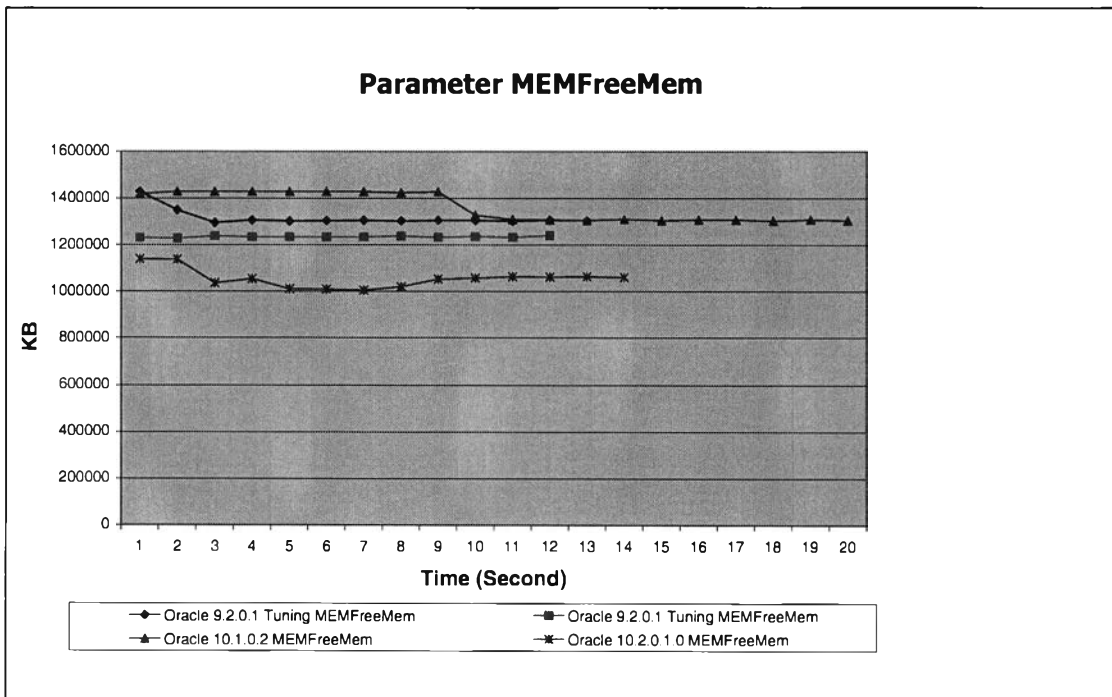
รูปที่ ค.3 แสดงการเปรียบเทียบพารามิเตอร์ CPUUtil ในการสร้างตาราง



รูปที่ ค.4 แสดงการเปรียบเทียบพารามิเตอร์ DSKReadWrite ในการสร้างตาราง

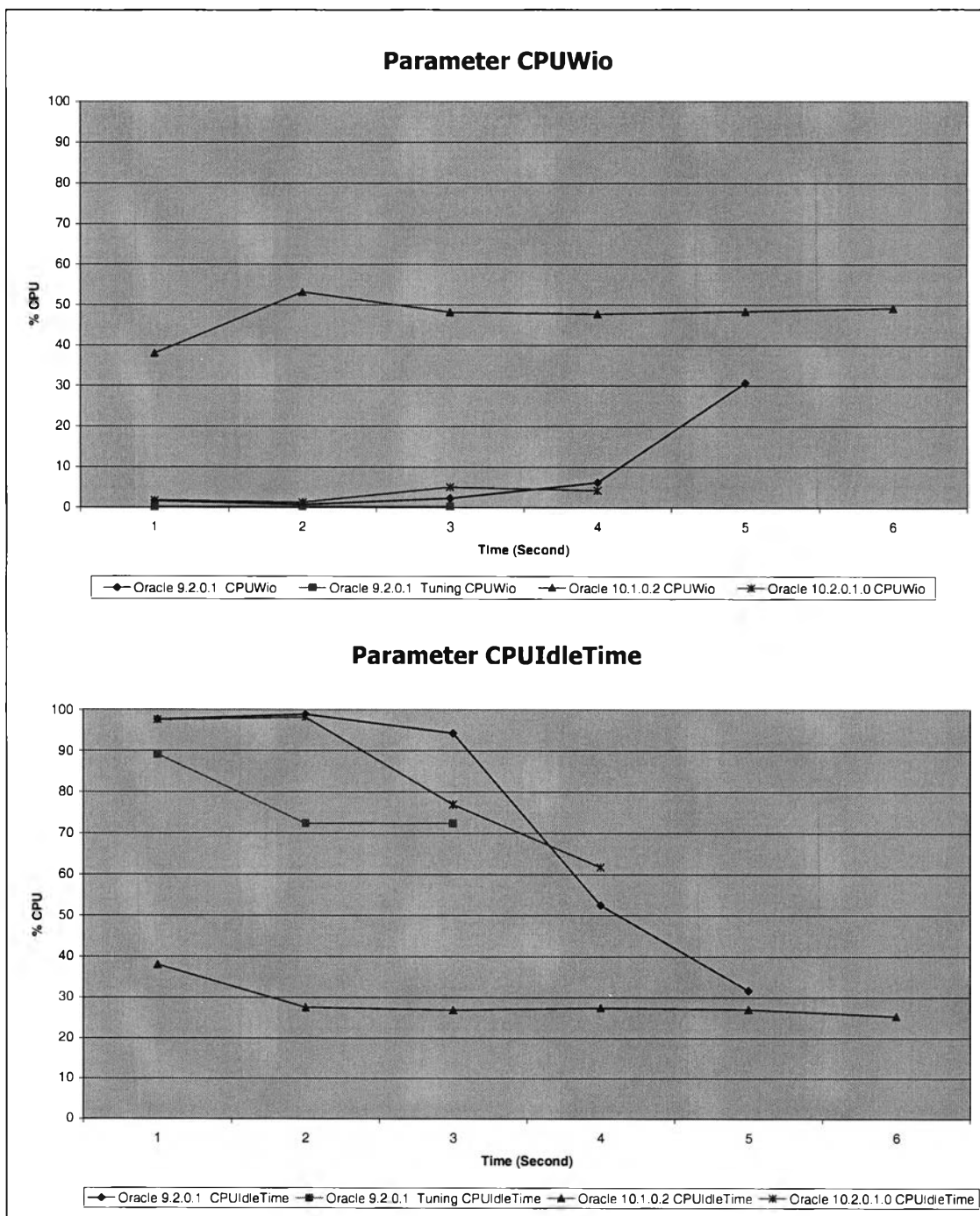


รูปที่ ค.5 แสดงการเปรียบเทียบพารามิเตอร์ DSKPresentBusy ในการสร้างตาราง

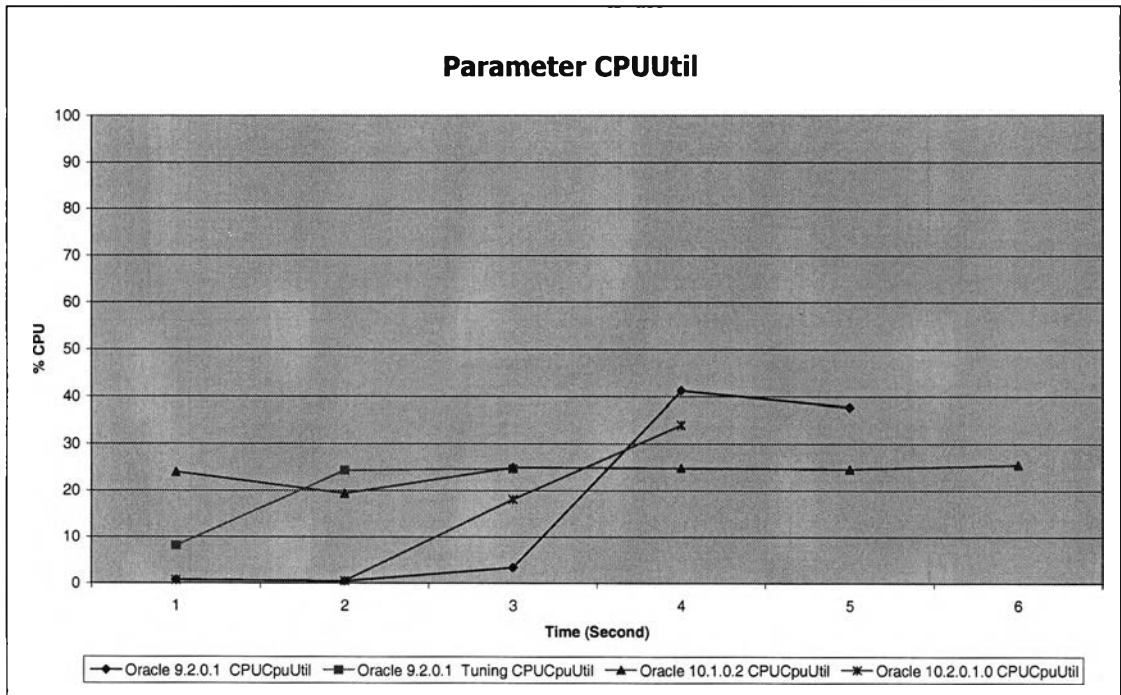


รูปที่ ค.6 แสดงการเปรียบเทียบพารามิเตอร์ DSKPresentBusyในการสร้างตาราง

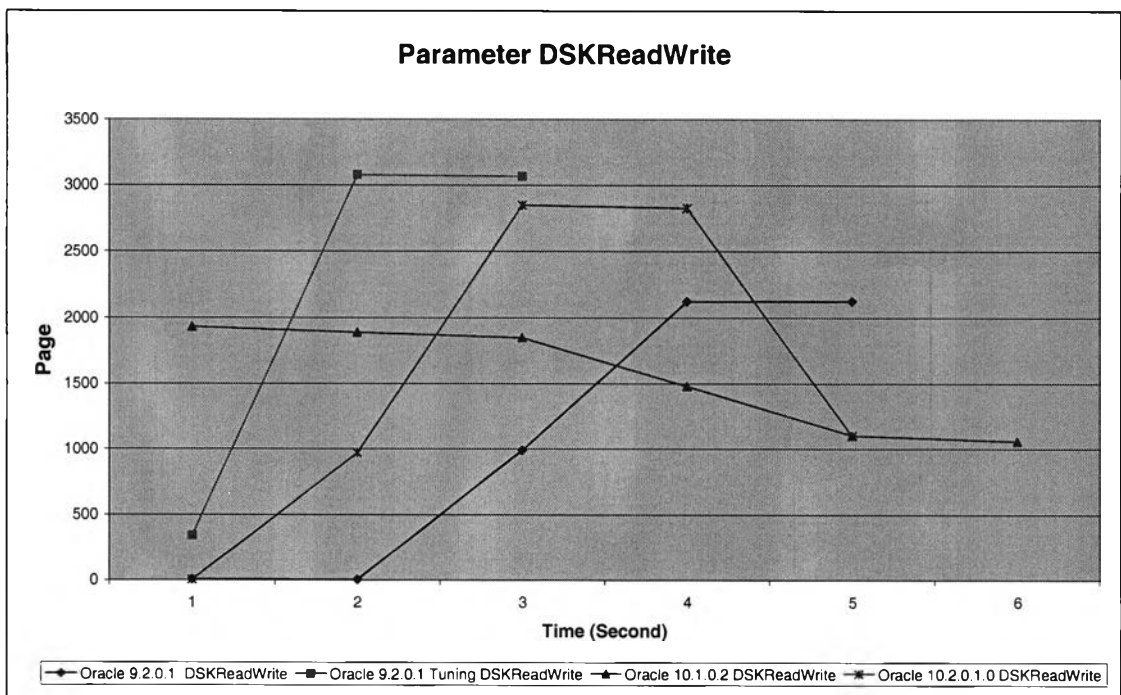
2) การเรียกดูข้อมูล มีการทดสอบการเรียกดูข้อมูล 2 แบบ คือการเรียกดูแบบ การดูทั้งหมด และการเรียกดูข้อมูลโดยใช้ parallel ผลปรากฏว่าในการเรียกดูในแบบที่ 2 นั้น การทำงานเร็วมากจึงไม่สามารถที่แสดงผลในรูปของกราฟได้ จึงนำเสนอเฉพาะในส่วนแบบที่ 1 เท่านั้น



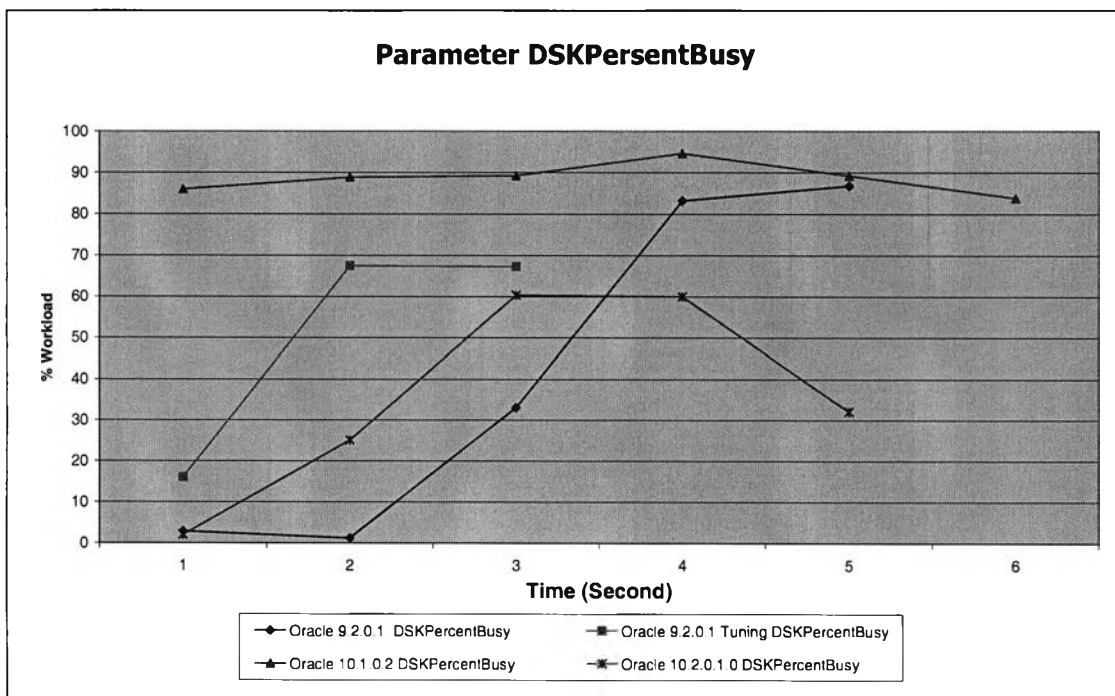
รูปที่ ค.7 แสดงการเปรียบเทียบพารามิเตอร์ CPUWio และ CPUIdleTime ในการเรียกดูข้อมูล



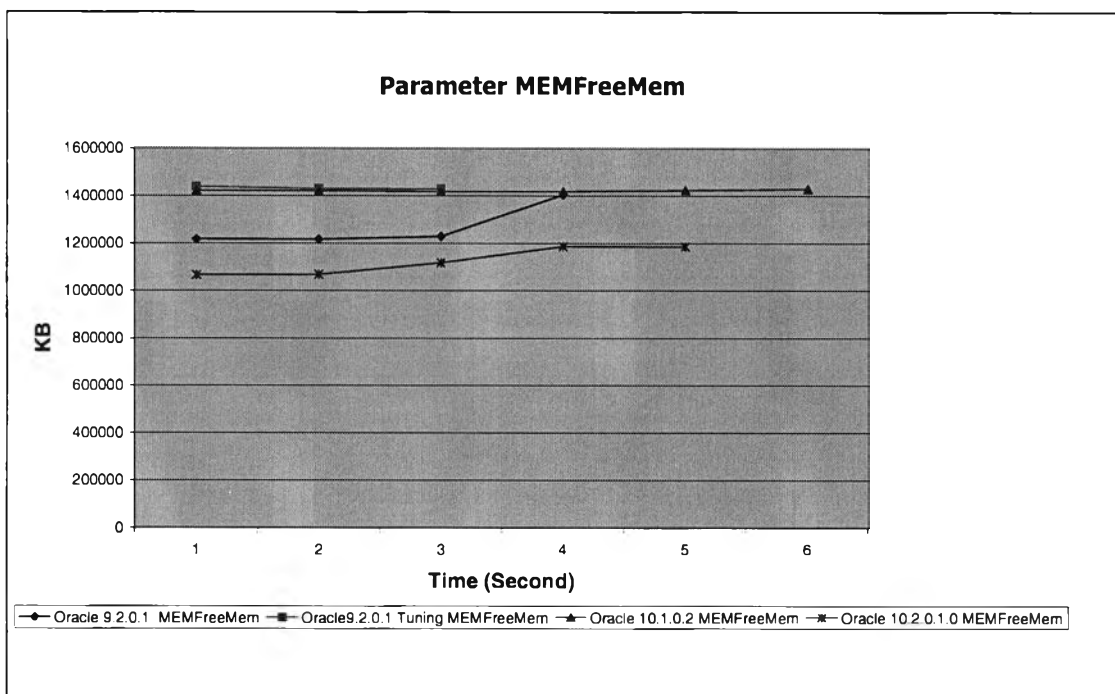
รูปที่ ค.8 แสดงการเปรียบเทียบพารามิเตอร์ CPUUtil ในการเรียกดูข้อมูล



รูปที่ ค.9 แสดงการเปรียบเทียบพารามิเตอร์ CPUUtil ในการเรียกดูข้อมูล

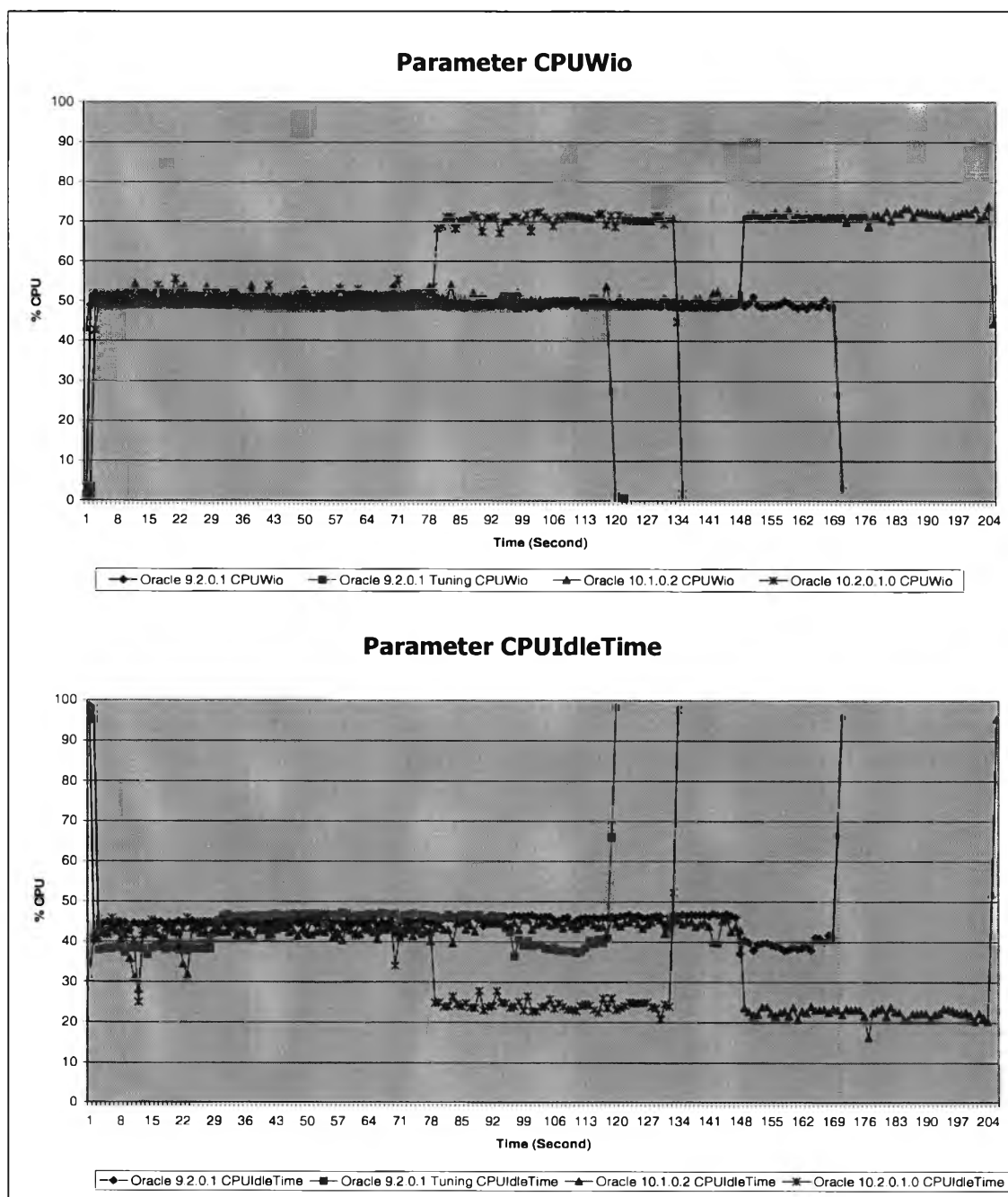


รูปที่ ค.10 แสดงการเปรียบเทียบพารามิเตอร์ DSKPercentBusy ในการเรียกดูข้อมูล

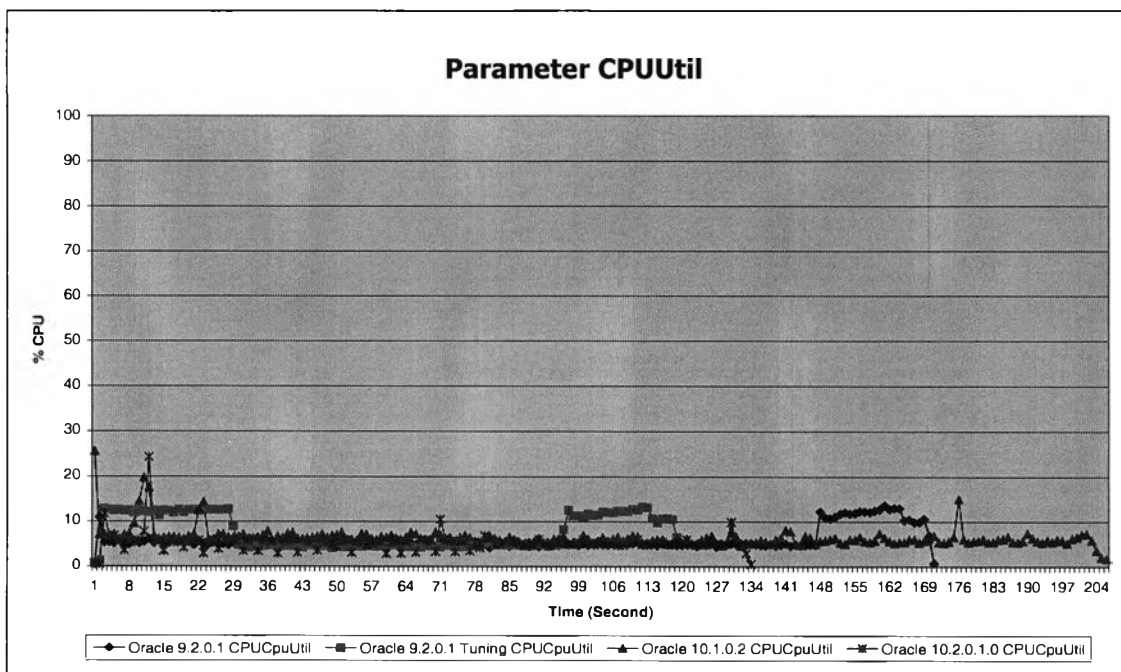


รูปที่ ค.11 แสดงการเปรียบเทียบพารามิเตอร์ MEMFreeMem ในการเรียกดูข้อมูล

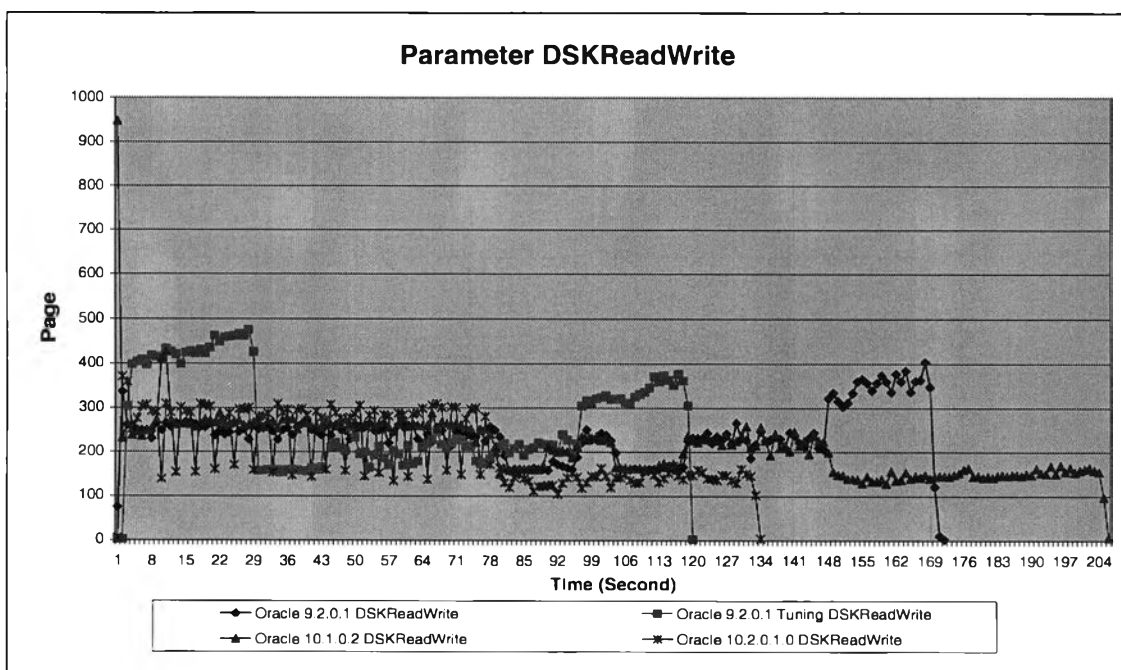
3) การสร้างอินเด็กซ์ ระบบจัดการฐานข้อมูลอราเคิล นั้นมีลักษณะเป็นแบบ B-Tree จากการทดสอบจะเห็นว่าอราเคิลเวอร์ชัน 10.2.0.1 นั้นมีการจัดการในเรื่องของการใช้ทรัพยากรได้ดี แต่อราเคิลเวอร์ชัน 9.2.0.1 ที่มีการเพิ่มประสิทธิภาพนั้น จะมีการใช้งานของซีพียู และดิสก์ที่น้อยกว่า ก็สามารถที่จะให้ผลดีเช่นกัน



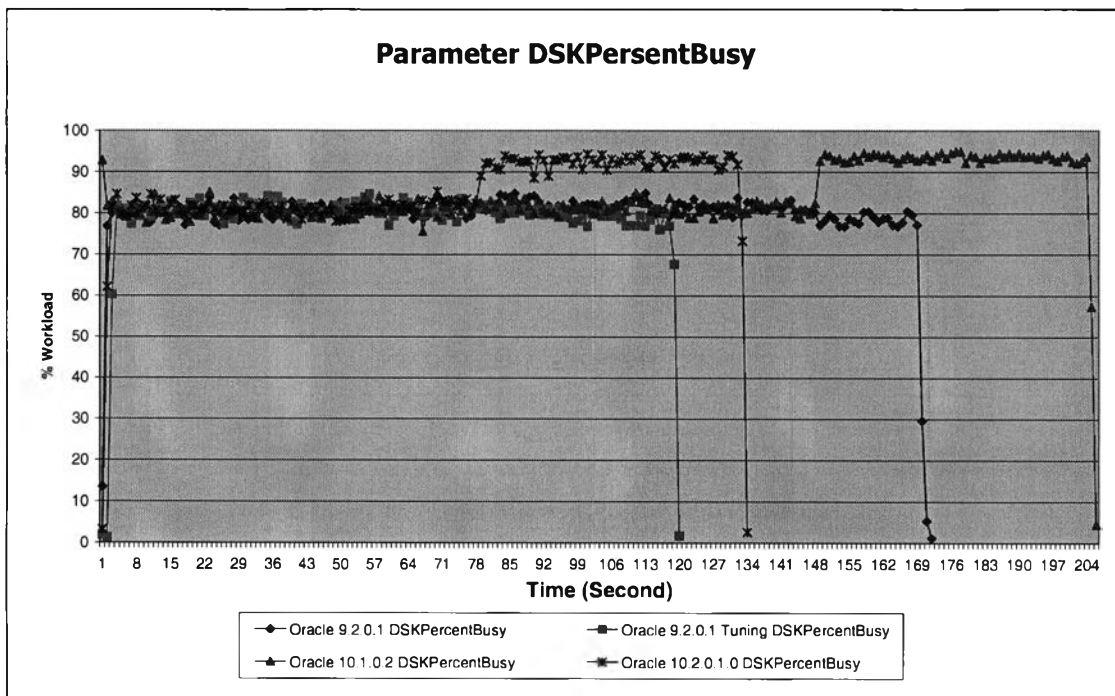
รูปที่ ค.12 แสดงการเปรียบเทียบพารามิเตอร์ CPUWio และ CPUIdleTime ในการสร้างอินเด็กซ์



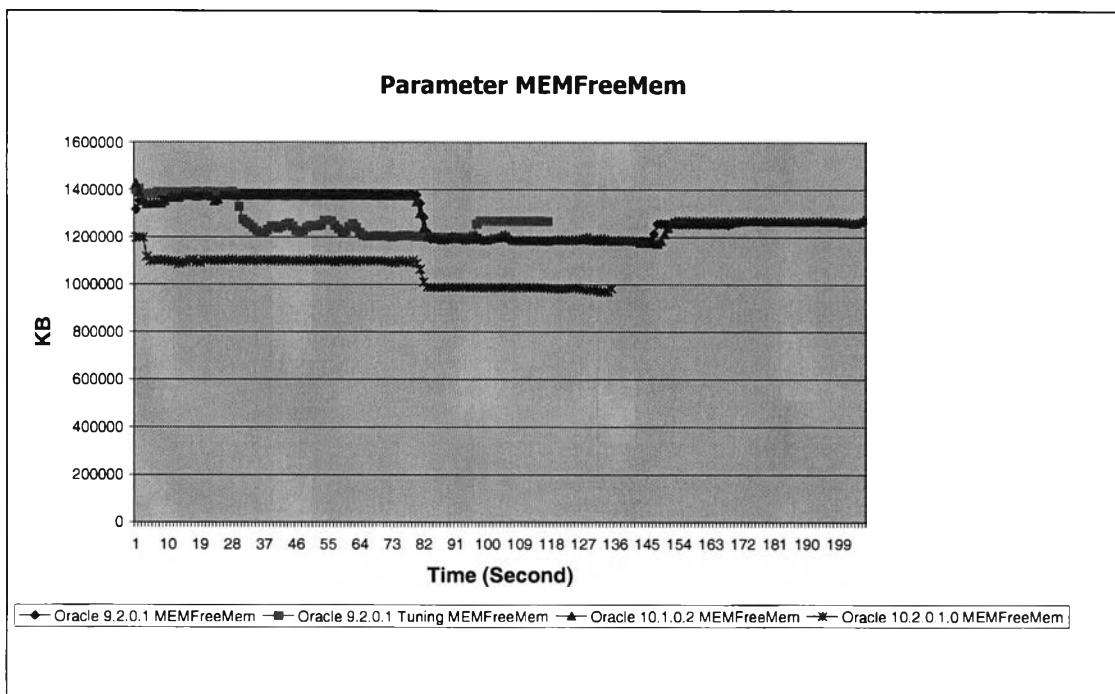
รูปที่ ค.13 แสดงการเปรียบเทียบพารามิเตอร์ CPUUtil ในการสร้างอินเด็กซ์



รูปที่ ค.14 แสดงการเปรียบเทียบพารามิเตอร์ DSKReadWrite ในการสร้างอินเด็กซ์

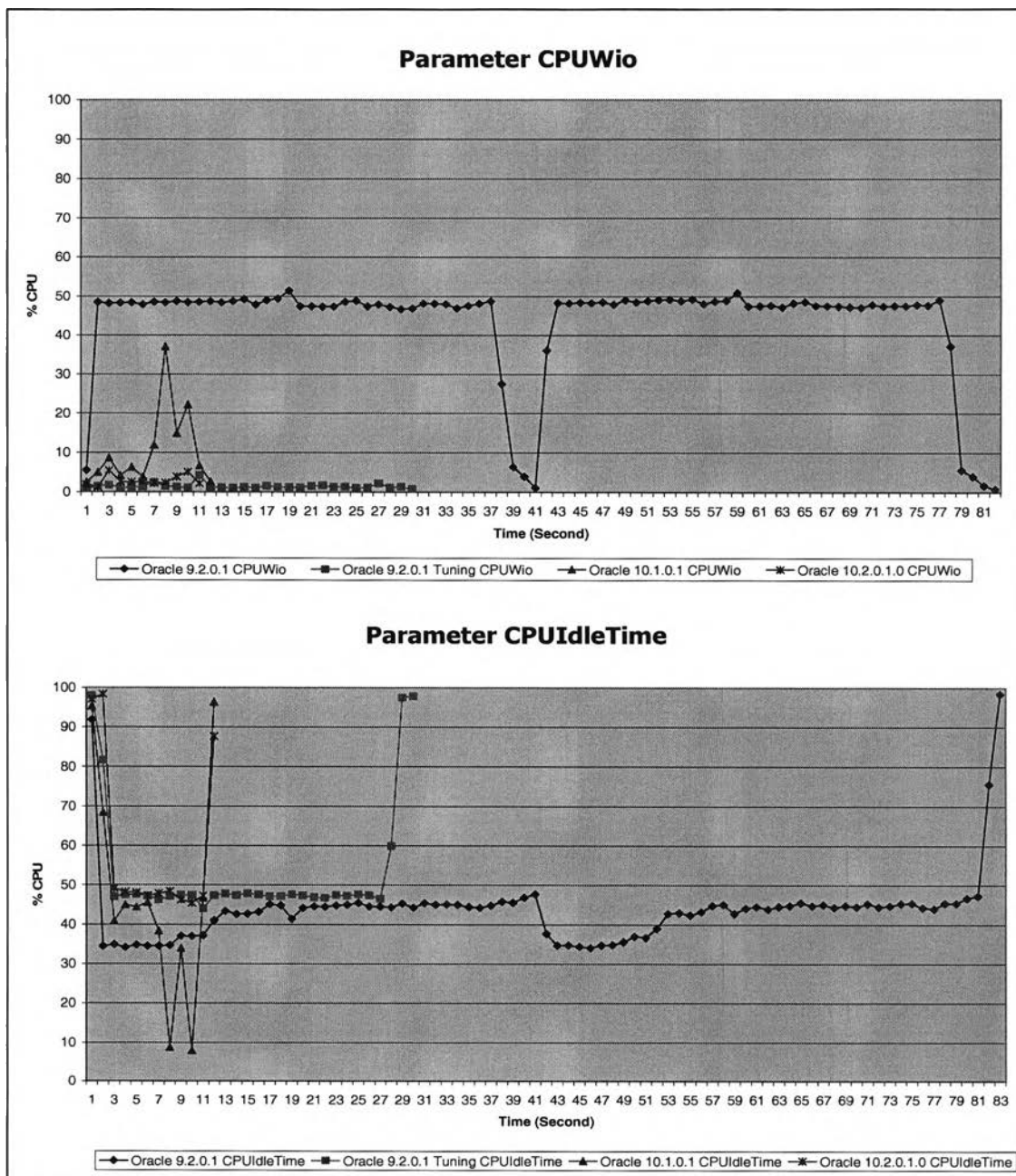


รูปที่ ค.15 แสดงการเปรียบเทียบพารามิเตอร์ DSKPercentBusy ในการสร้างอินเด็กซ์

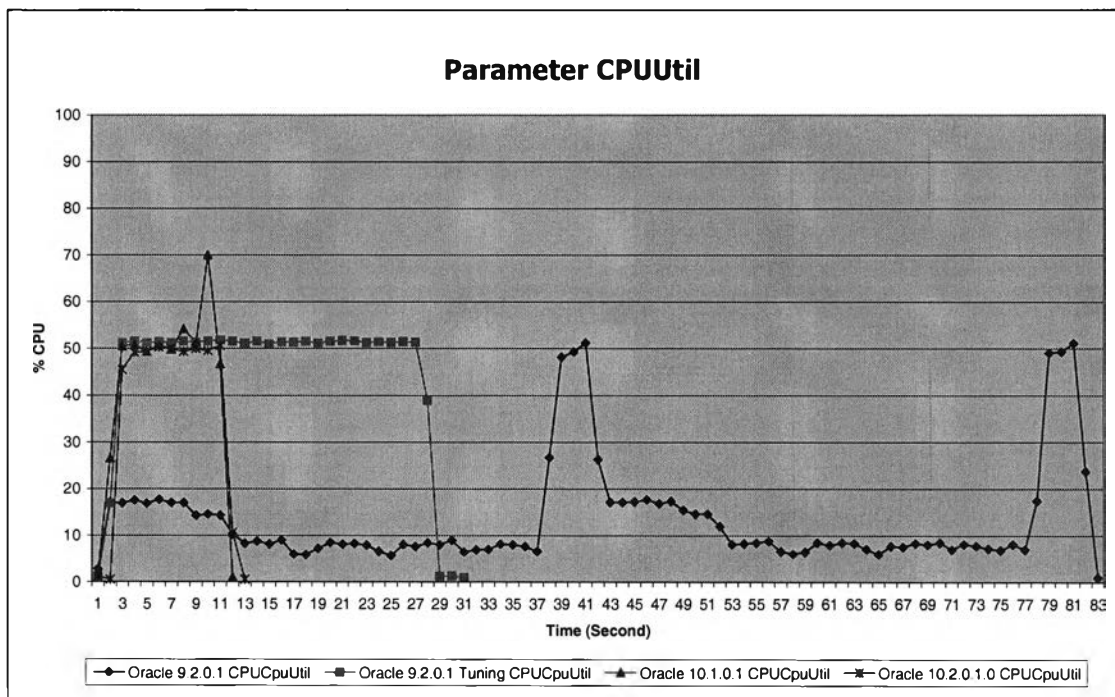


รูปที่ ค.16 แสดงการเปรียบเทียบพารามิเตอร์ DSKPercentBusy ในการสร้างอินเด็กซ์

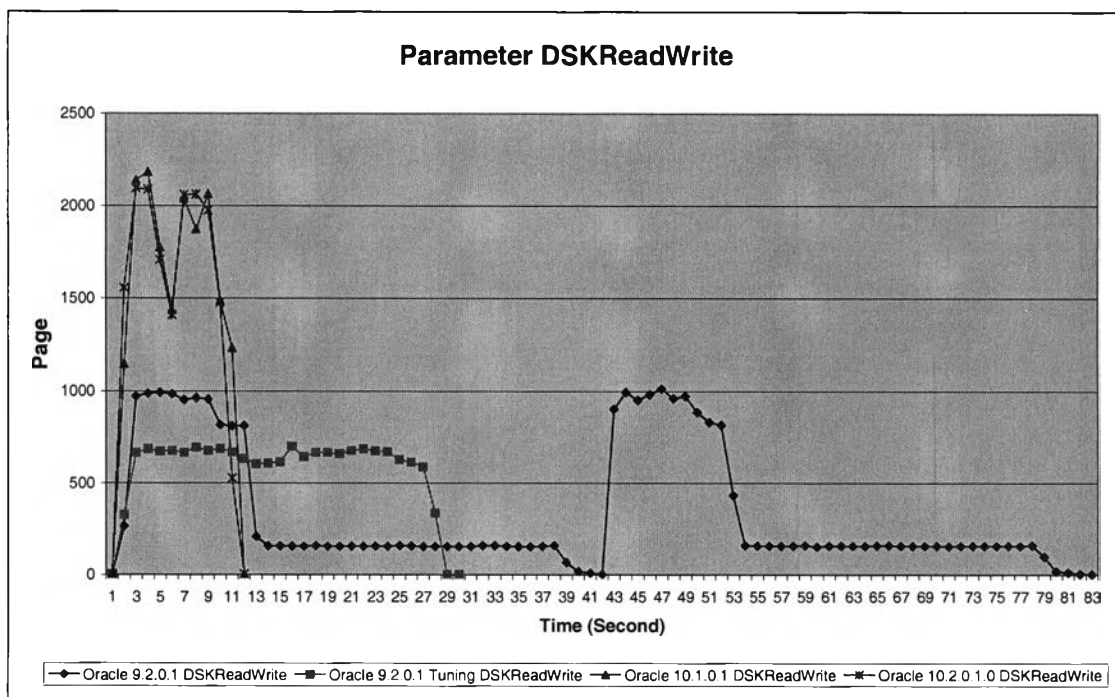
4) การเรียกดูข้อมูลโดยการใช้การเชื่อมต่อตาราง มีตารางที่ความสัมพันธ์กัน 2 ตาราง โดยตารางแรก คือ รายละเอียดการใช้งานโทรศัพท์เคลื่อนที่ มีจำนวนข้อมูลทั้งหมด 59,892,195 ระเบียบ ตารางที่สองเป็นรายละเอียดของประเภทบริการ เช่น ใช้โทรศัพท์ในการฟังเพลง หรือ ใช้เพื่อโทรหาหมายเลขปลายทางมีจำนวน 13,942 ระเบียบ



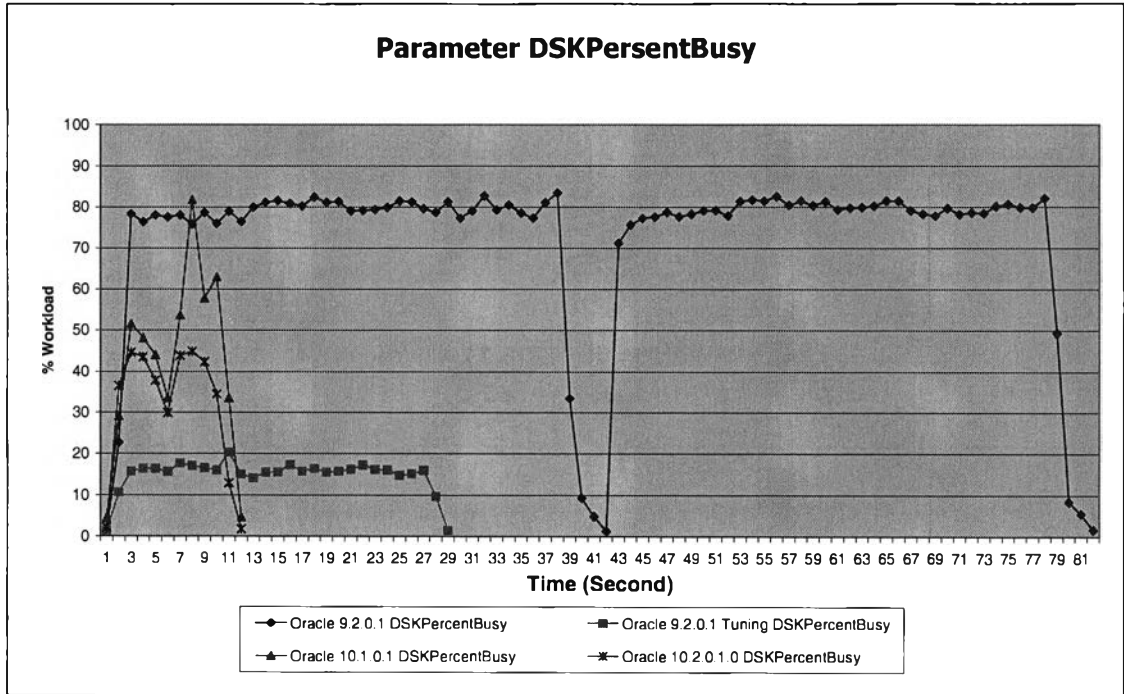
รูปที่ ค.17 แสดงการเปรียบเทียบพารามิเตอร์ CPUWio และ CPUIdleTime ในการ Join



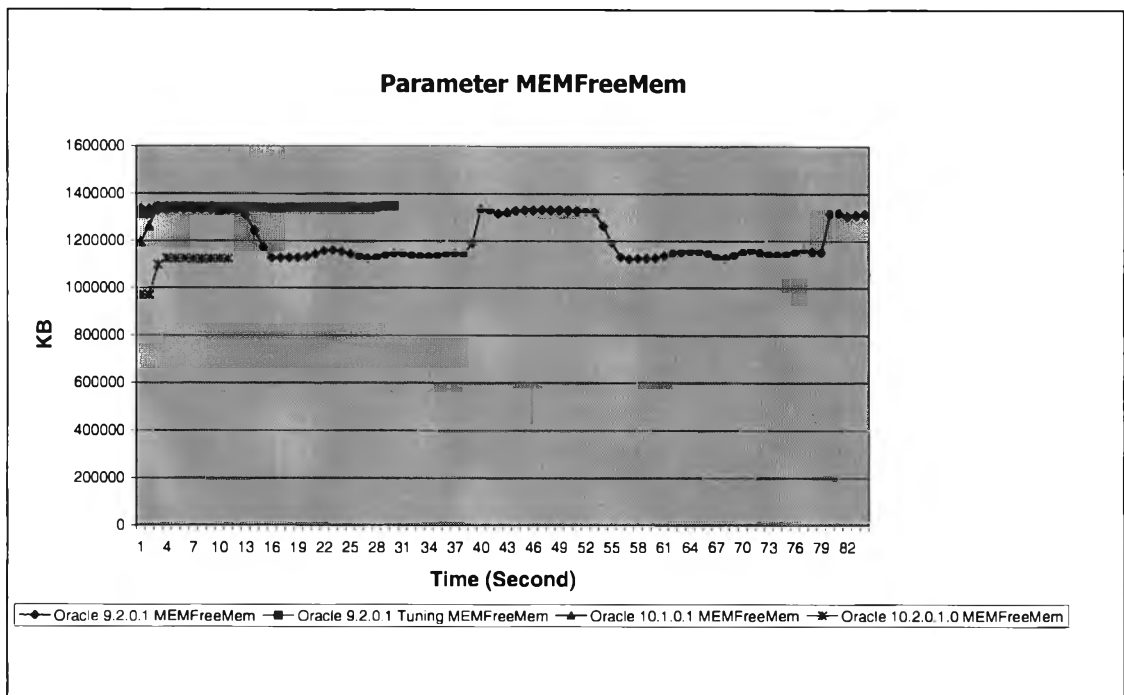
รูปที่ ค.18 แสดงการเปรียบเทียบพารามิเตอร์ CPUUtil ในการ Join



รูปที่ ค.19 แสดงการเปรียบเทียบพารามิเตอร์ DSKReadWrite ในการ Join

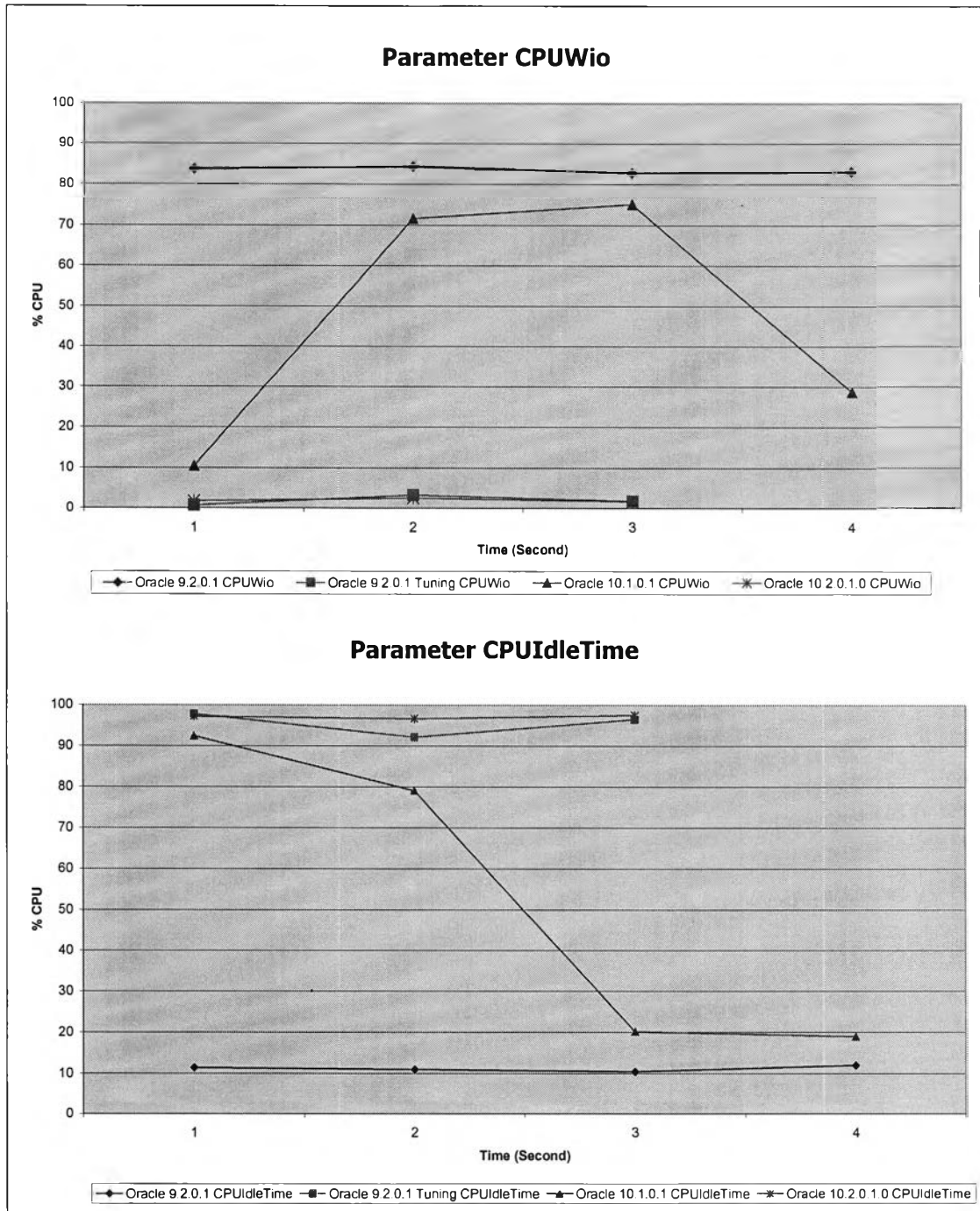


รูปที่ ค.20 แสดงการเปรียบเทียบพารามิเตอร์ DSKPercentBusy ในการ Join

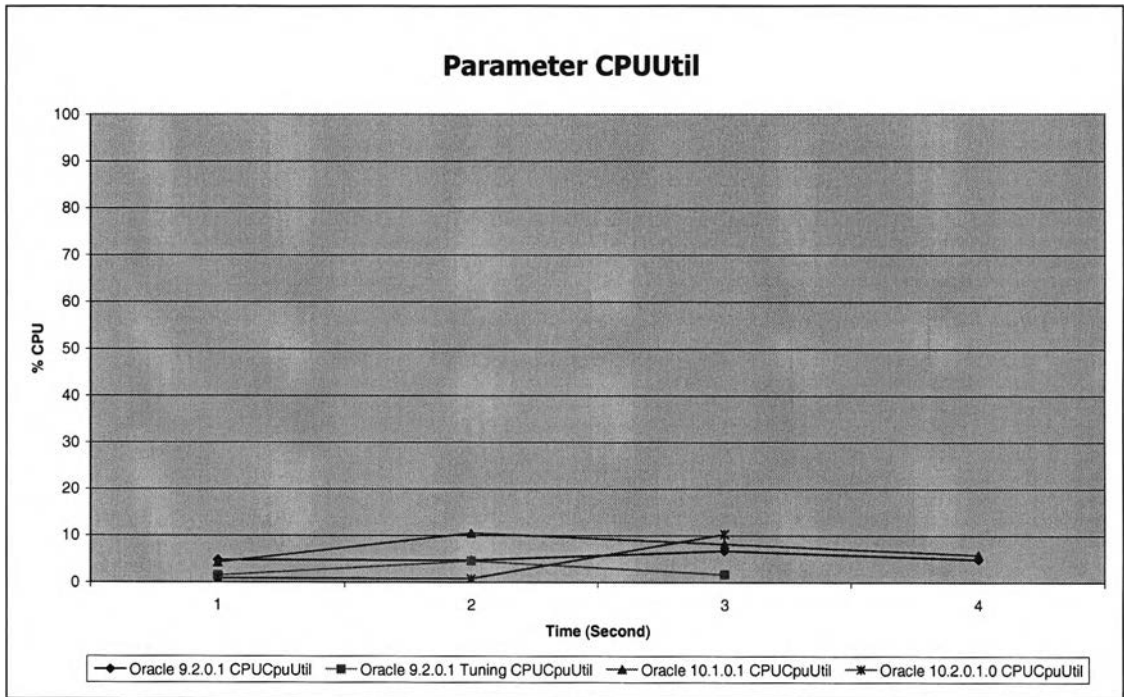


รูปที่ ค.21 แสดงการเปรียบเทียบพารามิเตอร์ MEMFreeMem ในการ Join

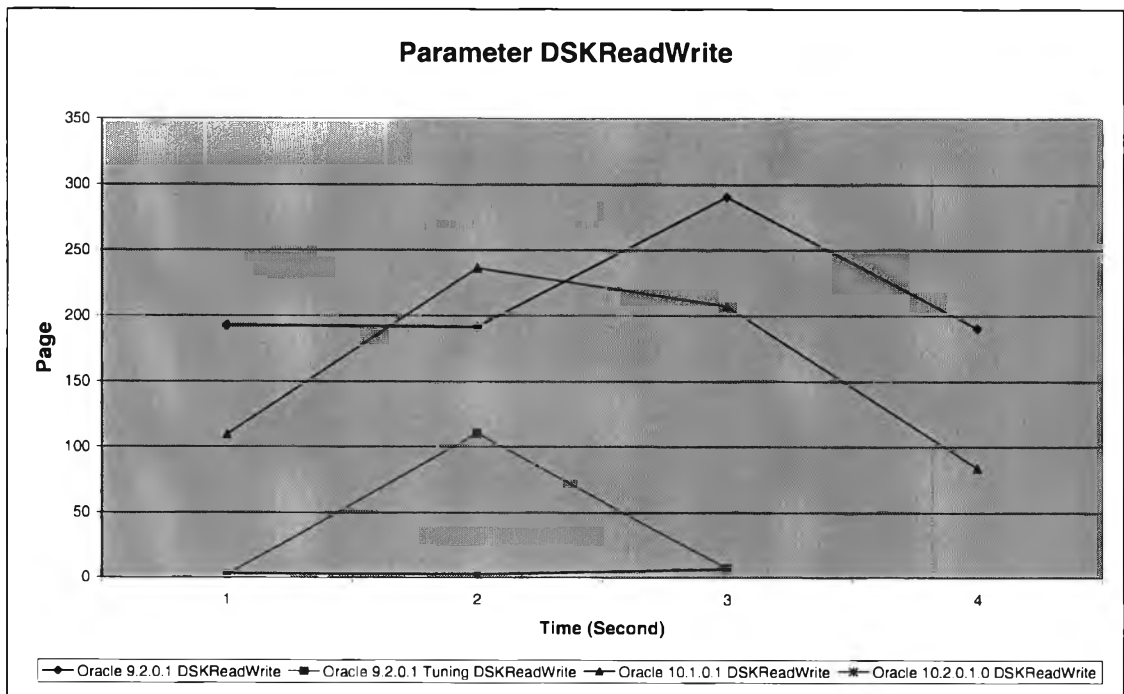
5) การเพิ่มข้อมูลในปริมาณมาก เป็นการเพิ่มข้อมูลจำนวน 200,000 ระเบียบในตาราง ซึ่งผลการทำงานออราเคิลเวอร์ชัน 10.2.0.1 จะมีอัตราการใช้ทรัพยากรที่ใกล้เคียงกับออราเคิลเวอร์ชัน 9.2.0.1 ที่มีการเพิ่มประสิทธิภาพ



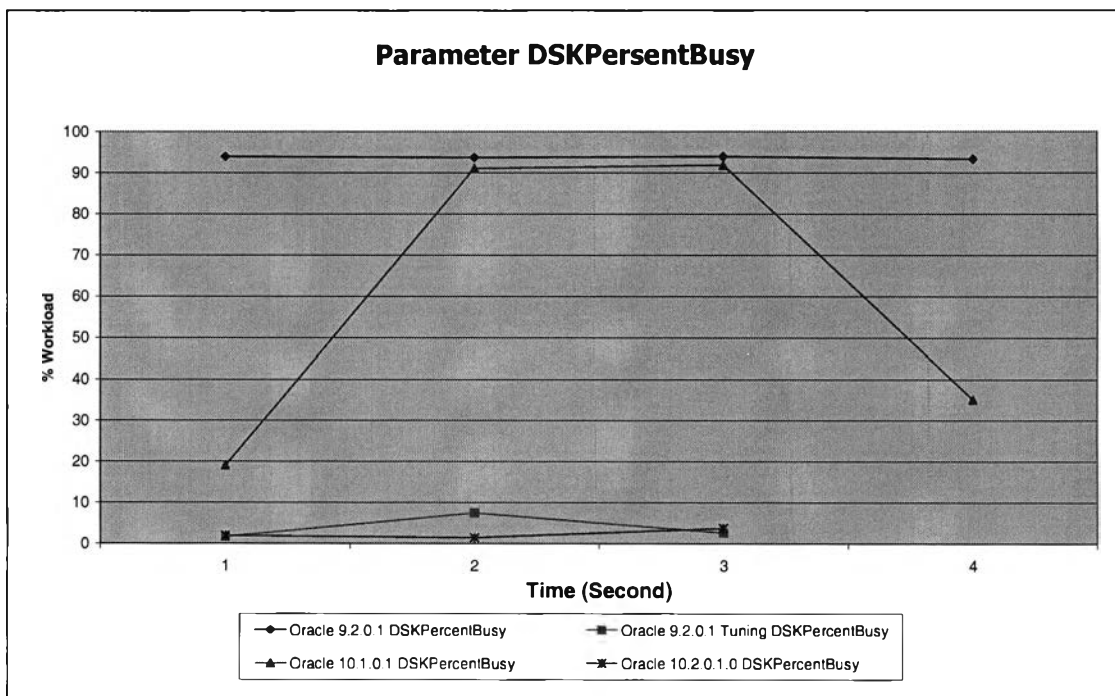
รูปที่ ค.22 แสดงการเปรียบเทียบพารามิเตอร์ CPUWio และ CPUIdleTime ในการ Insert



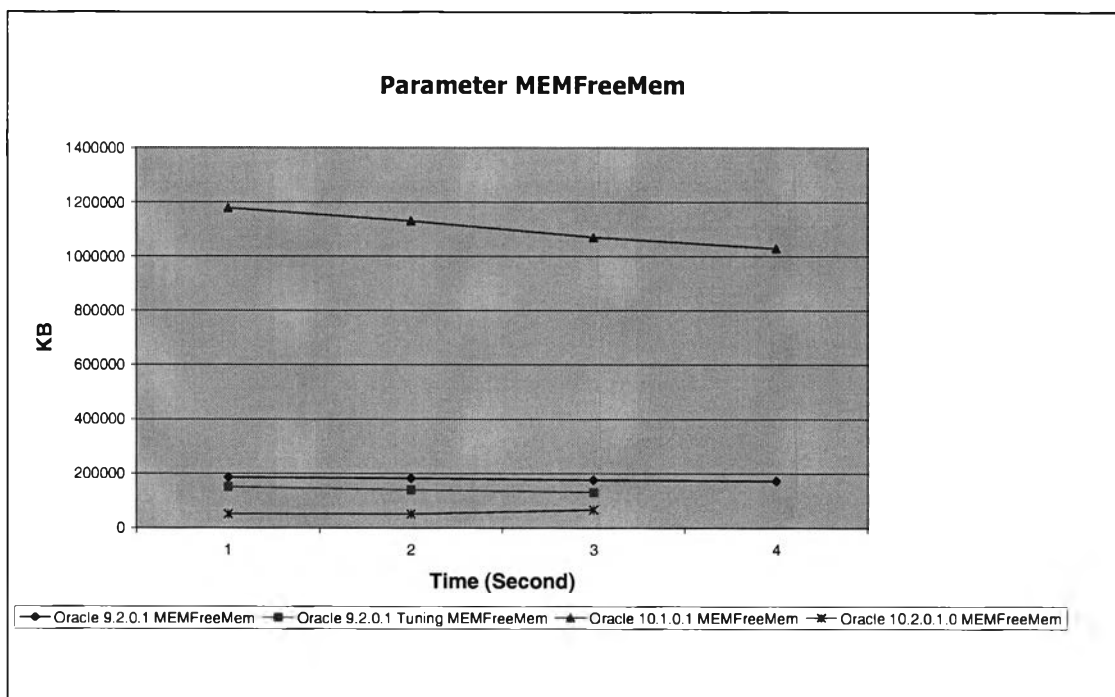
รูปที่ ค.23 แสดงการเปรียบเทียบพารามิเตอร์ CPUUtil ในการ Insert



รูปที่ ค.24 แสดงการเปรียบเทียบพารามิเตอร์ DSKReadWrite ในการ Insert

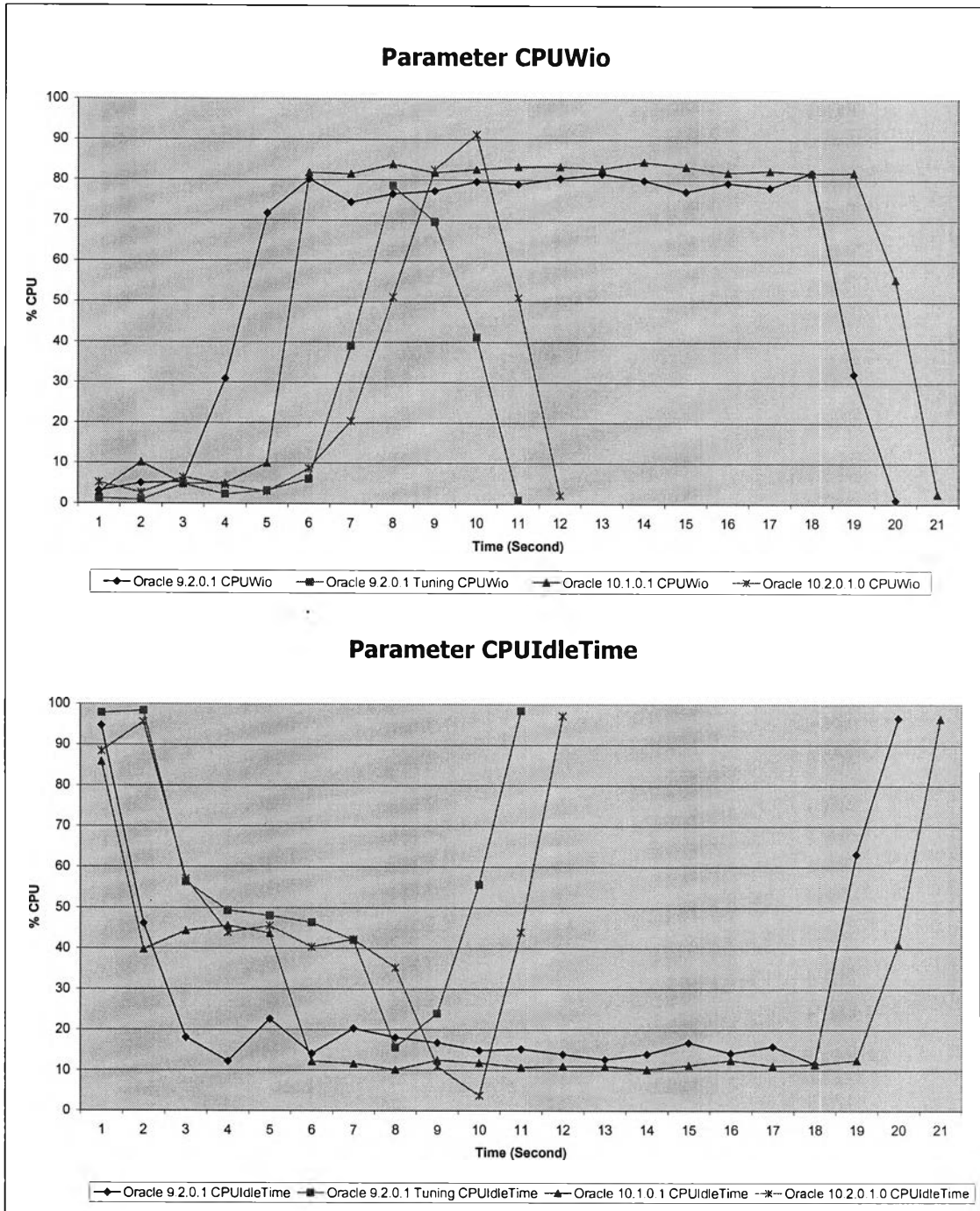


รูปที่ ค.25 แสดงการเปรียบเทียบพารามิเตอร์ DSKPercentBusy ในการ Insert

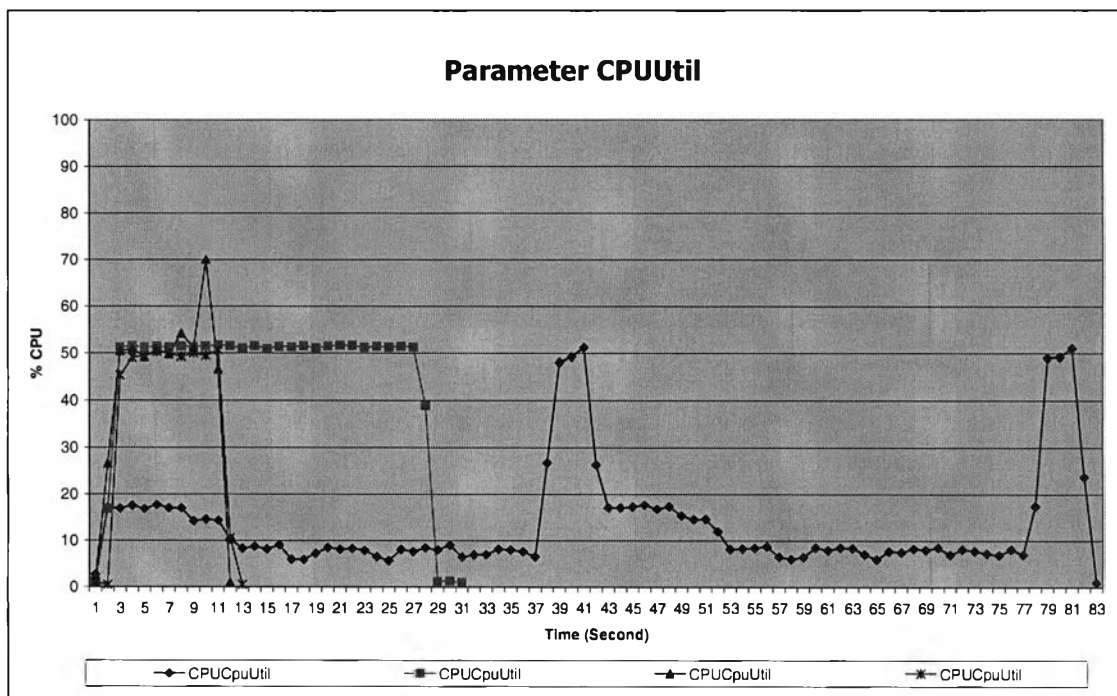


รูปที่ ค.26 แสดงการเปรียบเทียบพารามิเตอร์ MEMFreeMem ในการ Insert

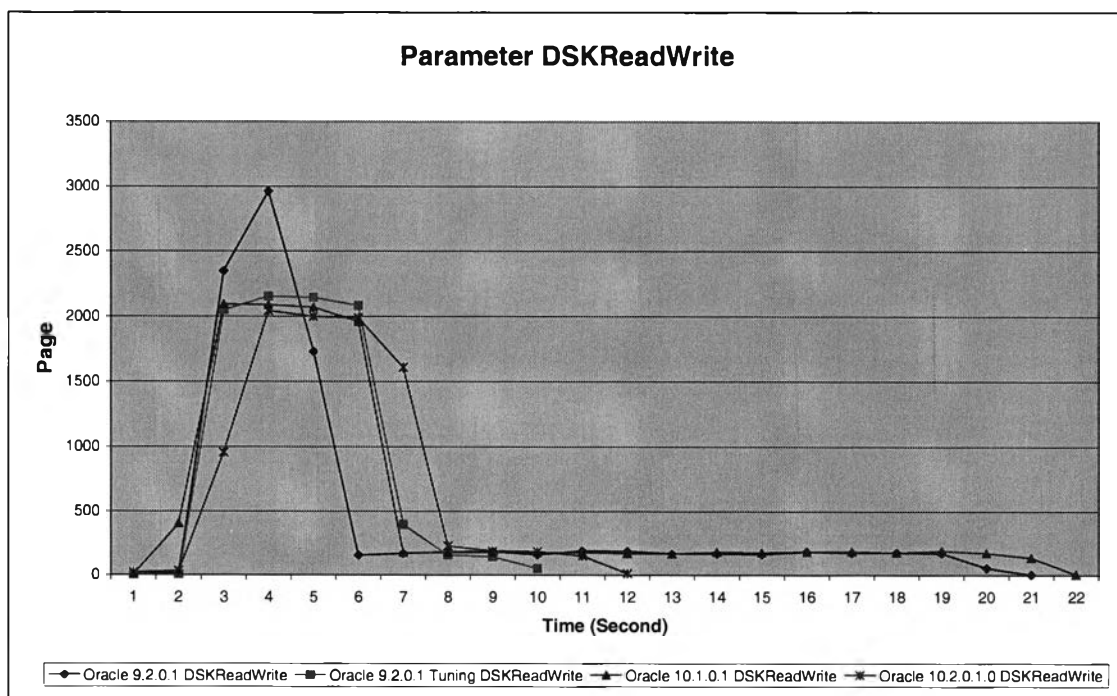
6) การลบข้อมูลออกจากตาราง เป็นการนำข้อมูลออกจากตารางเป็นจำนวน 200,000 ระเบียบัน ผลจะเห็นได้ว่าการใช้งานของของทรัพยากรของออราเคิลเวอร์ชัน 10.2.0.1 นั้น มีการจัดการได้ดีใกล้เคียงกับออราเคิลเวอร์ชัน 9.2.0.1 ที่มีการเพิ่มประสิทธิภาพ



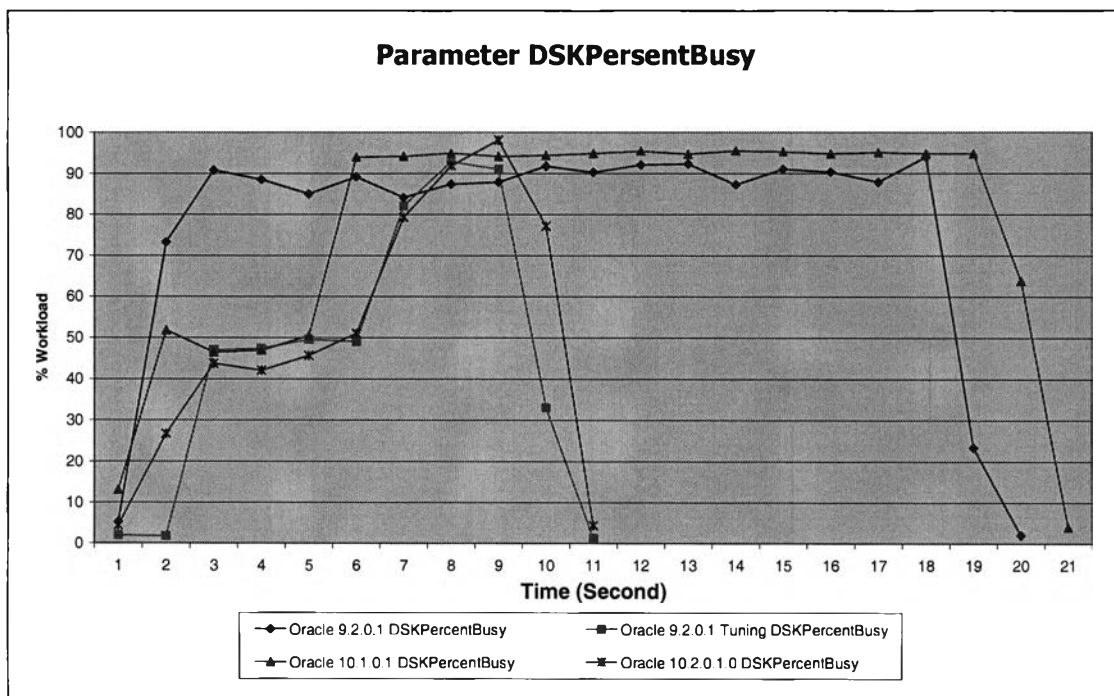
รูปที่ ค.28 แสดงการเปรียบเทียบพารามิเตอร์ CPUWio และ CPUIdleTime ในการ delete



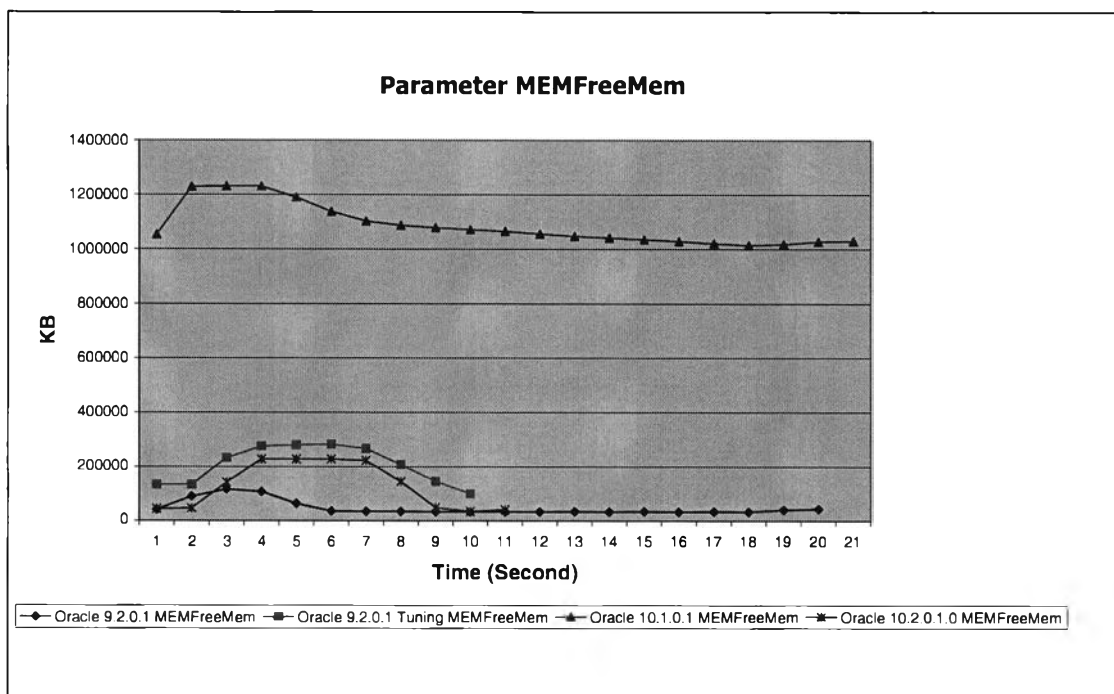
รูปที่ ค.29 แสดงการเปรียบเทียบพารามิเตอร์ CPUUtil ในการ delete



รูปที่ ค.30 แสดงการเปรียบเทียบพารามิเตอร์ DSKReadWrite ในการ delete

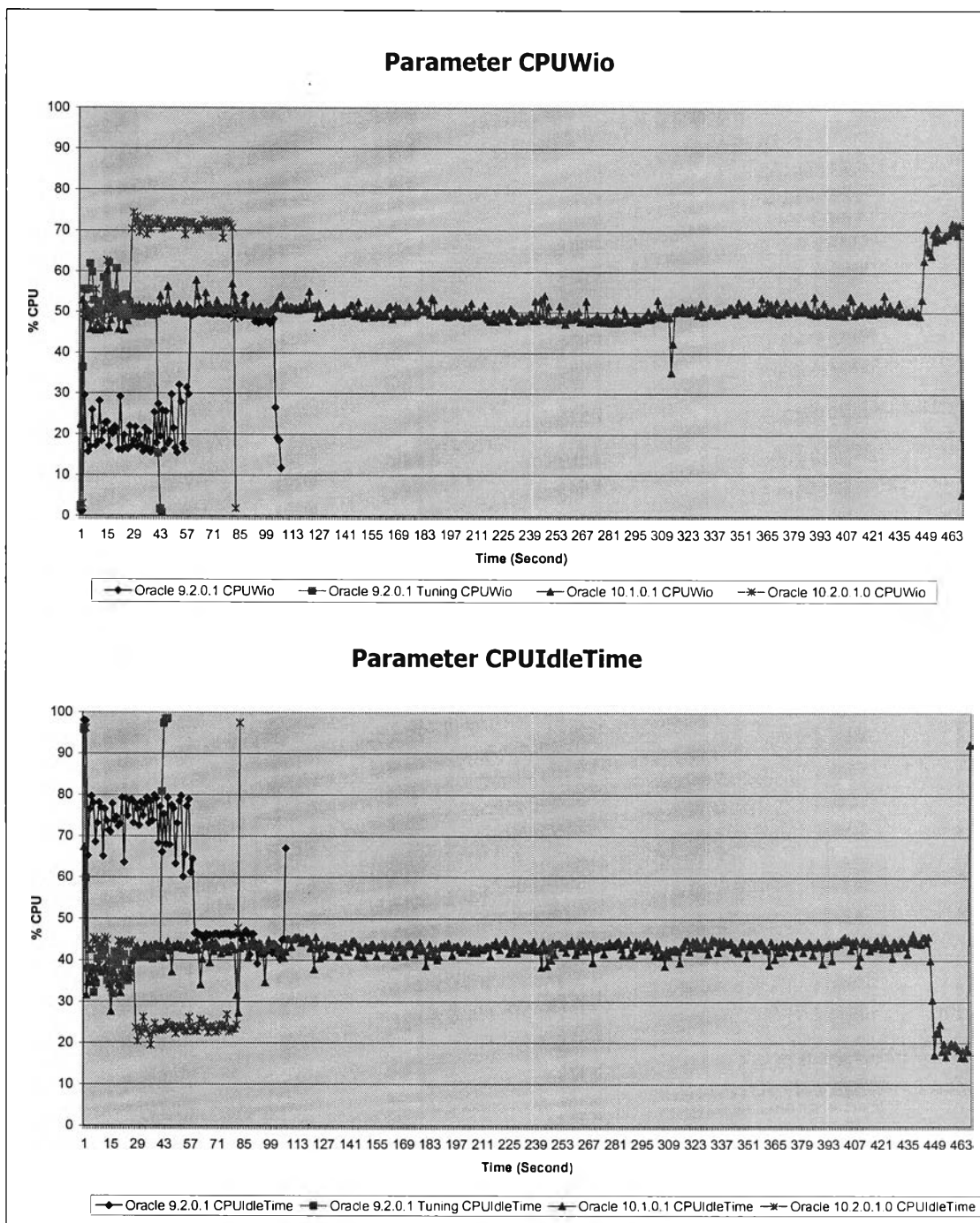


รูปที่ ค.31 แสดงการเปรียบเทียบพารามิเตอร์ DSKPercentBusy ในการ delete

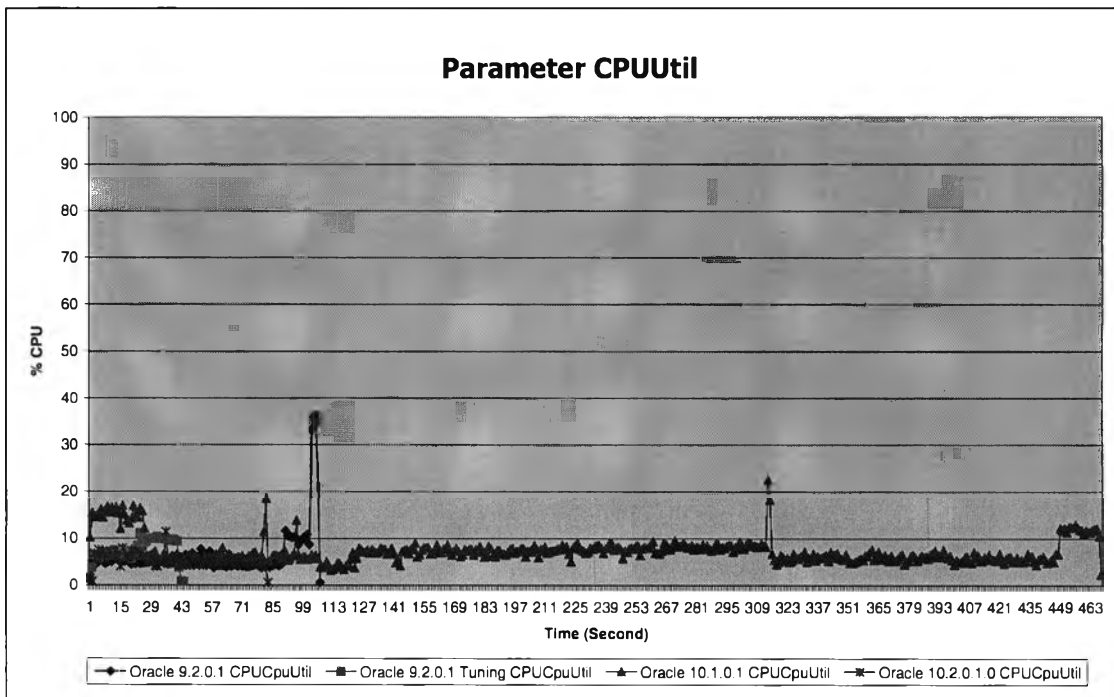


รูปที่ ค.32 แสดงการเปรียบเทียบพารามิเตอร์ MEMFreeMem ในการ delete

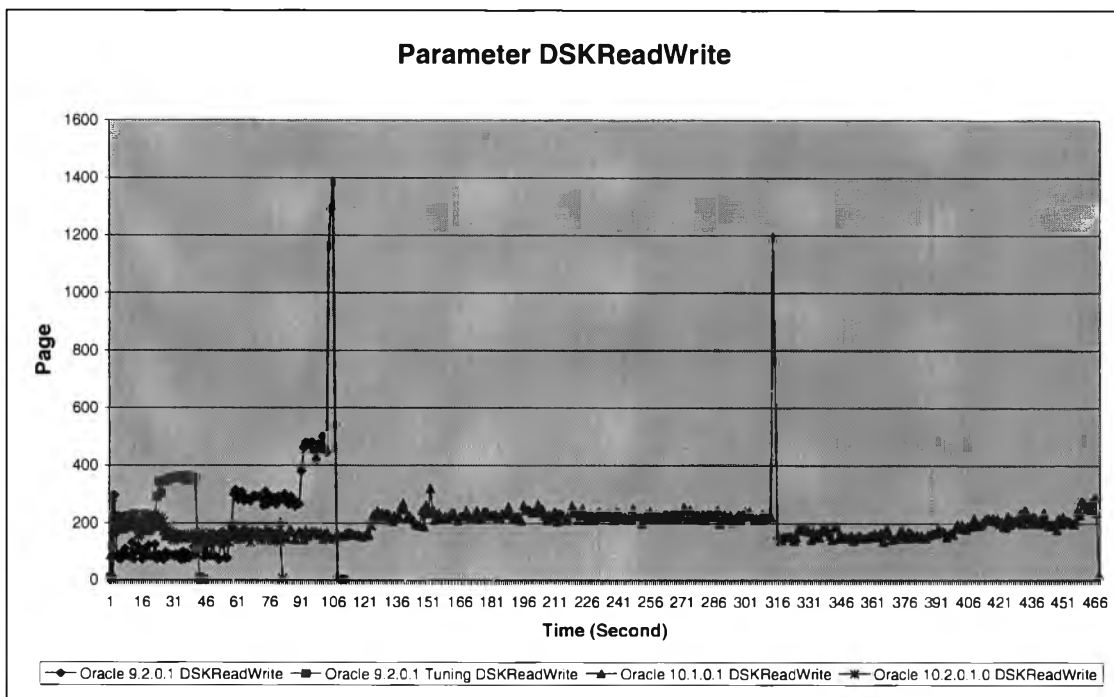
7) การปรับปรุงอินเด็กซ์ การบริหารการใช้ทรัพยากรของออราเคิลเวอร์ชัน 10.1.0.2 จะมากกว่าออราเคิลเวอร์ชันอื่นๆ ในทรัพยากรทุกตัวอย่างเห็นได้ชัด



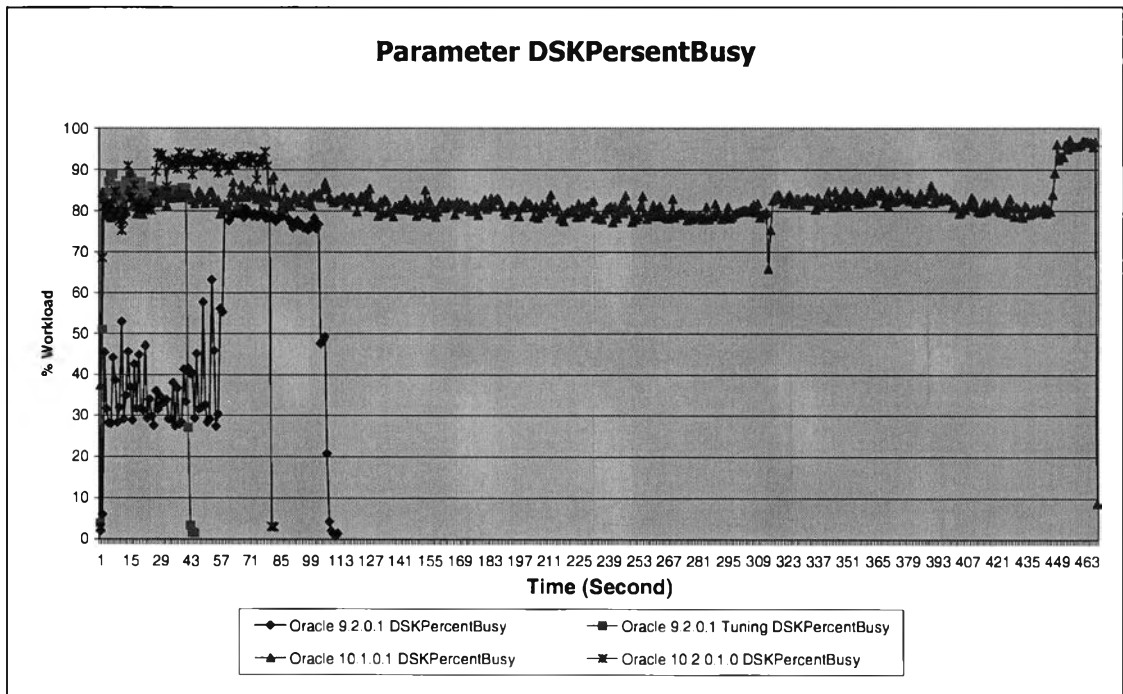
รูปที่ ค.33 แสดงการเปรียบเทียบพารามิเตอร์ CPUWio และ CPUIdleTime ในการ Rebuild index



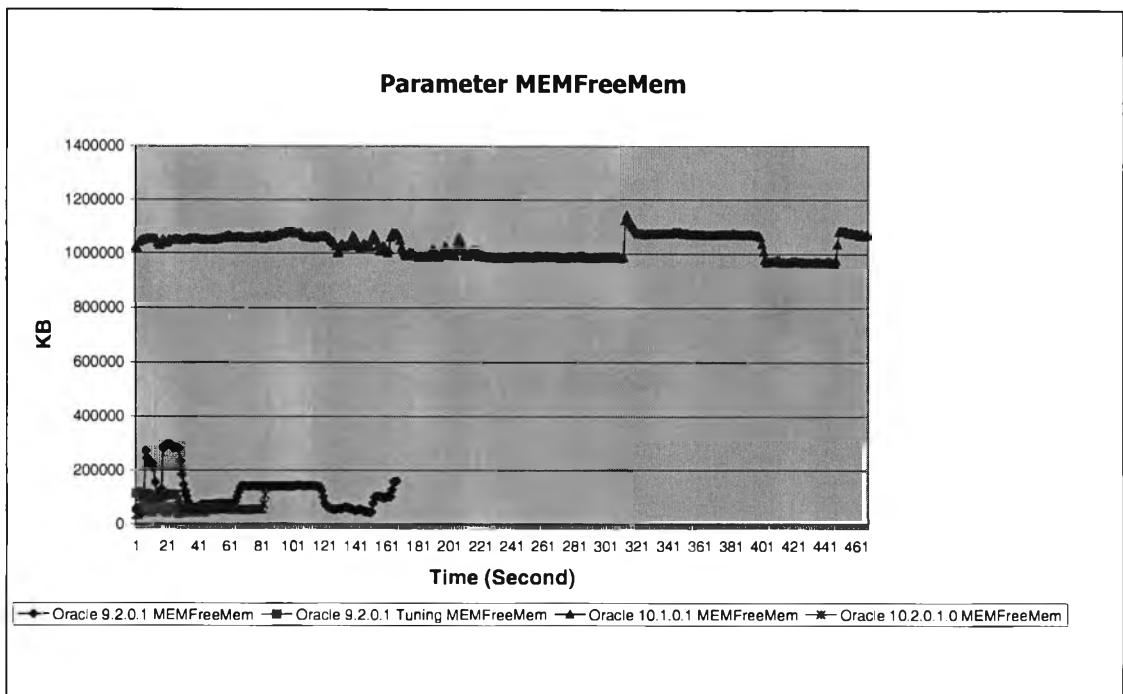
รูปที่ ค.34 แสดงการเปรียบเทียบพารามิเตอร์ CPUUtil ในการ Rebuild index



รูปที่ ค.35 แสดงการเปรียบเทียบพารามิเตอร์ DSKReadWrite ในการ Rebuild index

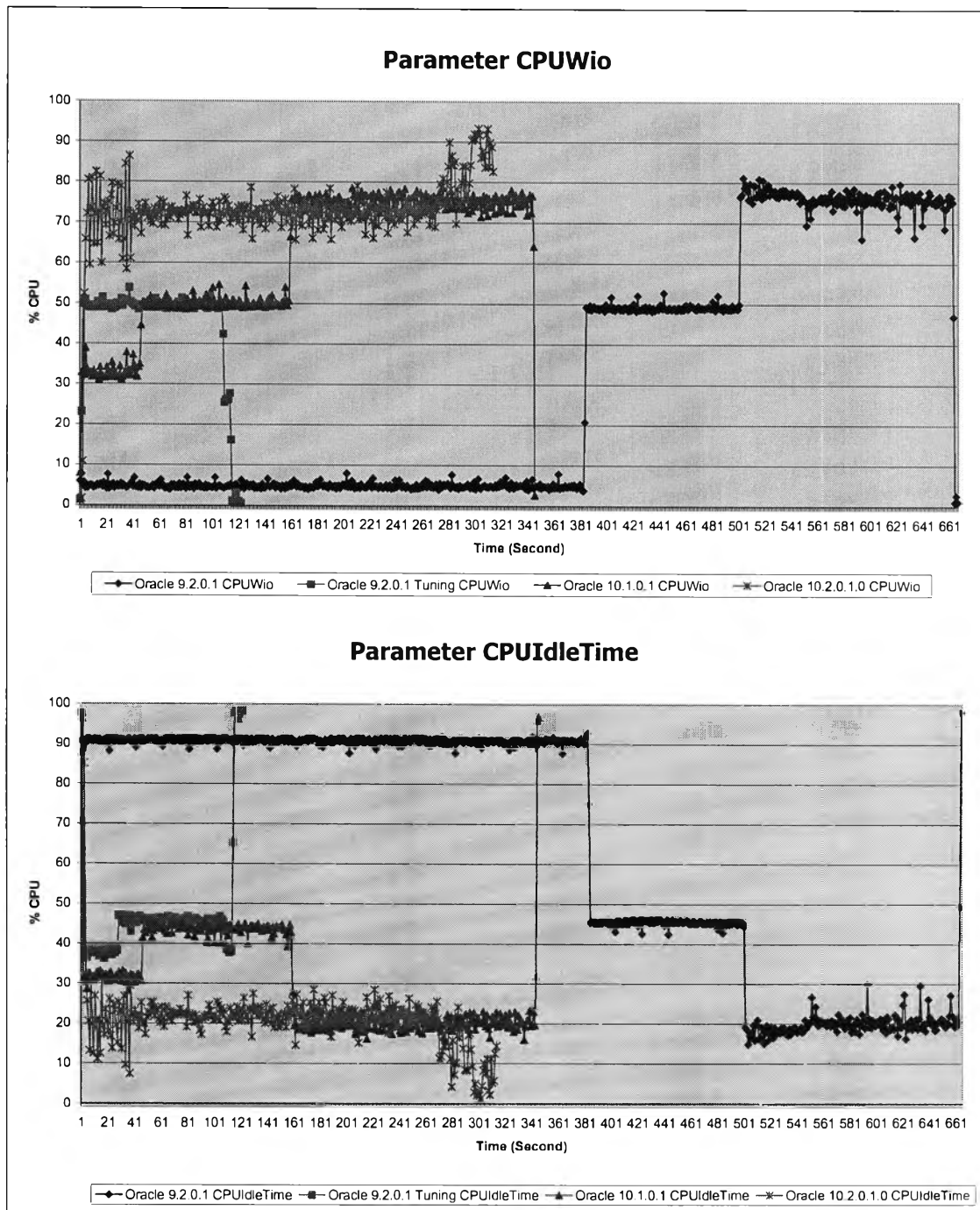


รูปที่ ค.36 แสดงการเปรียบเทียบพารามิเตอร์ DSKPercentBusy ในการ Rebuild index

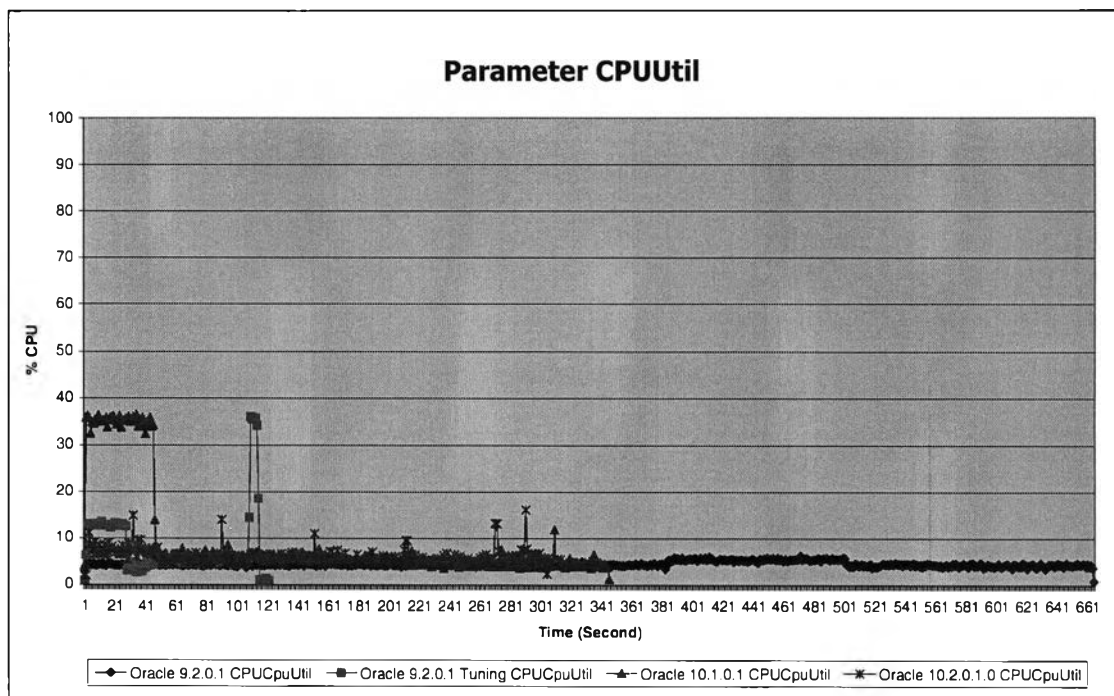


รูปที่ ค.37 แสดงการเปรียบเทียบพารามิเตอร์ MEMFreeMem ในการ Rebuild index

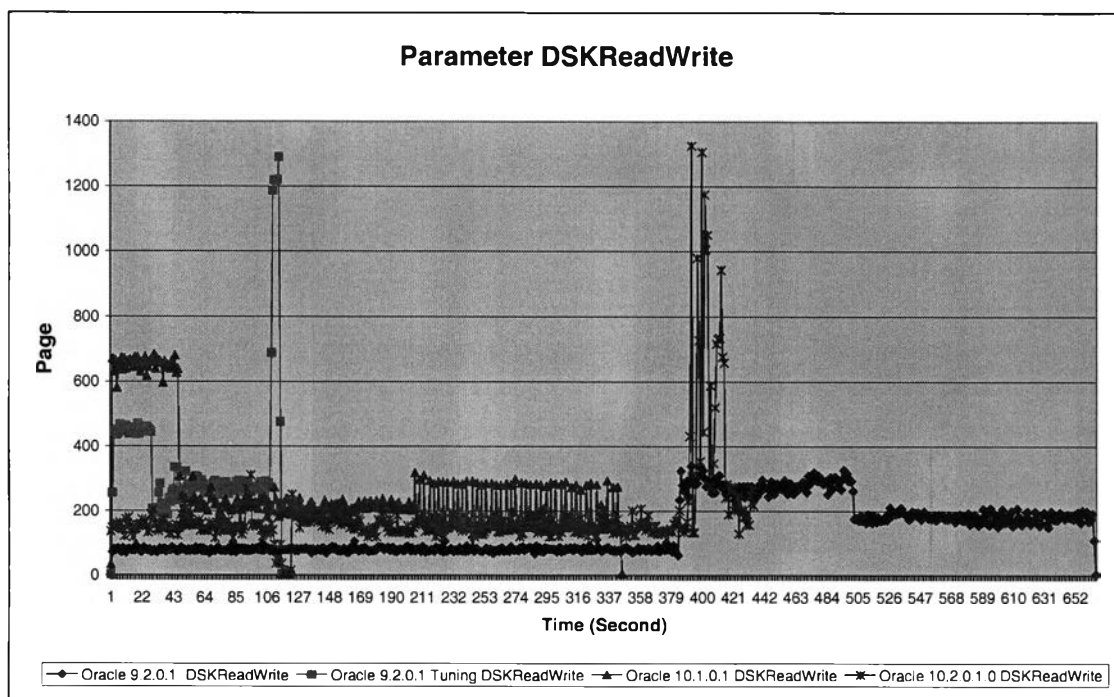
8) การนำข้อมูลจากตารางปกติ เข้าสู่ตารางที่เป็นพาดิชั่น ที่มีโครงสร้างที่ต่างกัน จะเห็นได้ว่าการบริหารทรัพยากรของออราเคิลเวอร์ชัน 9.2.0.1 จะดีกว่าออราเคิลเวอร์ชันอื่น ๆ



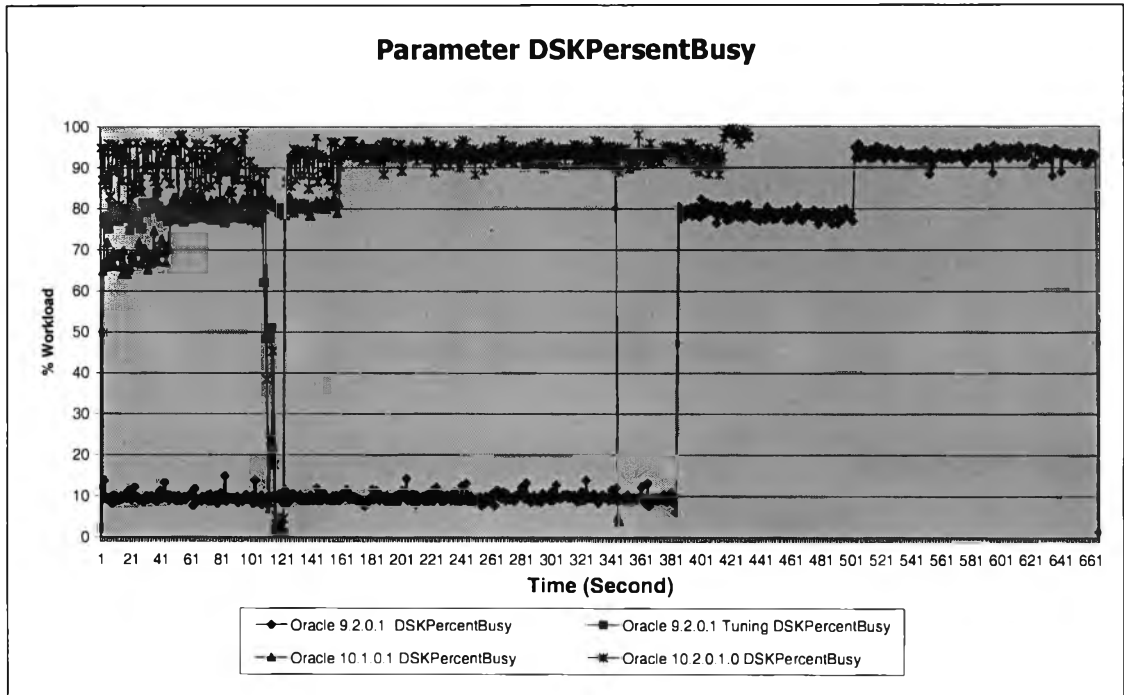
รูปที่ ค.38 แสดงการเปรียบเทียบพารามิเตอร์ CPUWio และ CPUIdleTime ในการ insert partition



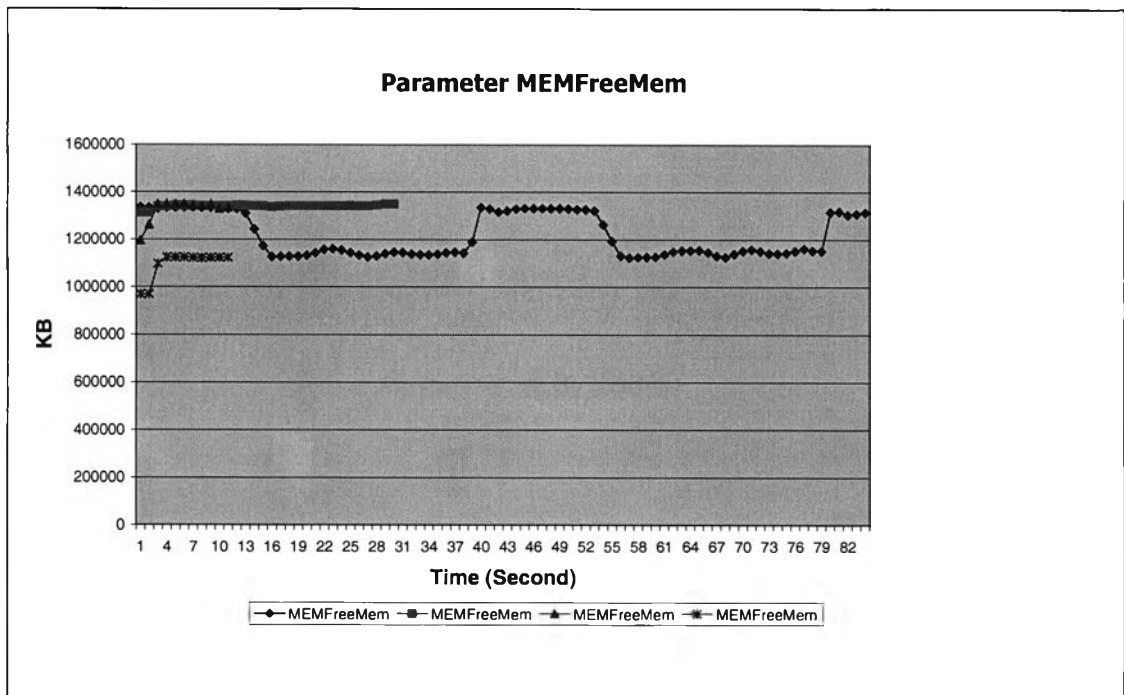
รูปที่ ค.39 แสดงการเปรียบเทียบพารามิเตอร์ CPUUtil ในการ insert partition



รูปที่ ค.40 แสดงการเปรียบเทียบพารามิเตอร์ DSKReadWrite ในการ insert partition

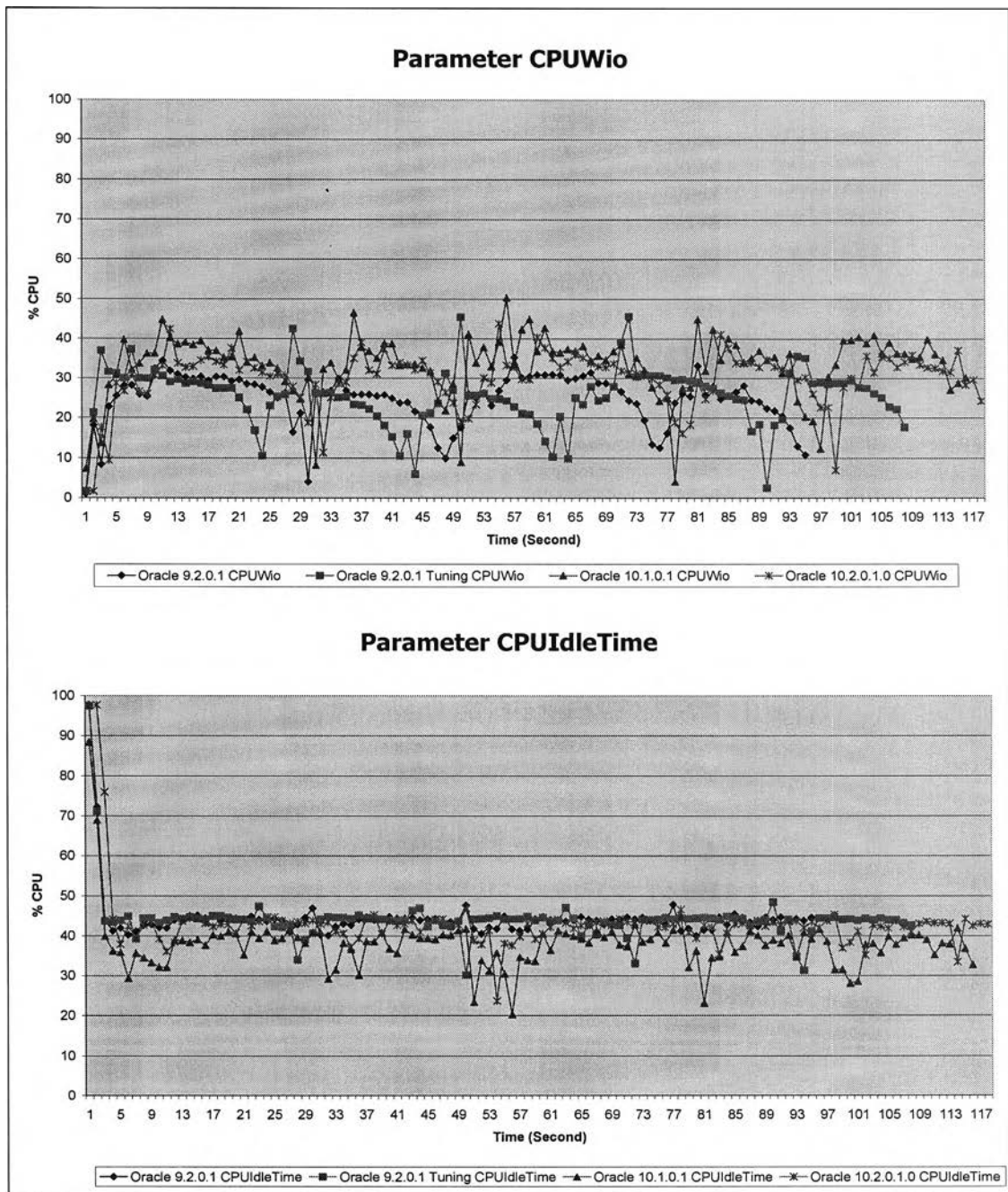


รูปที่ ค.41 แสดงการเปรียบเทียบพารามิเตอร์ DSKPercentBusy ในการ insert partition

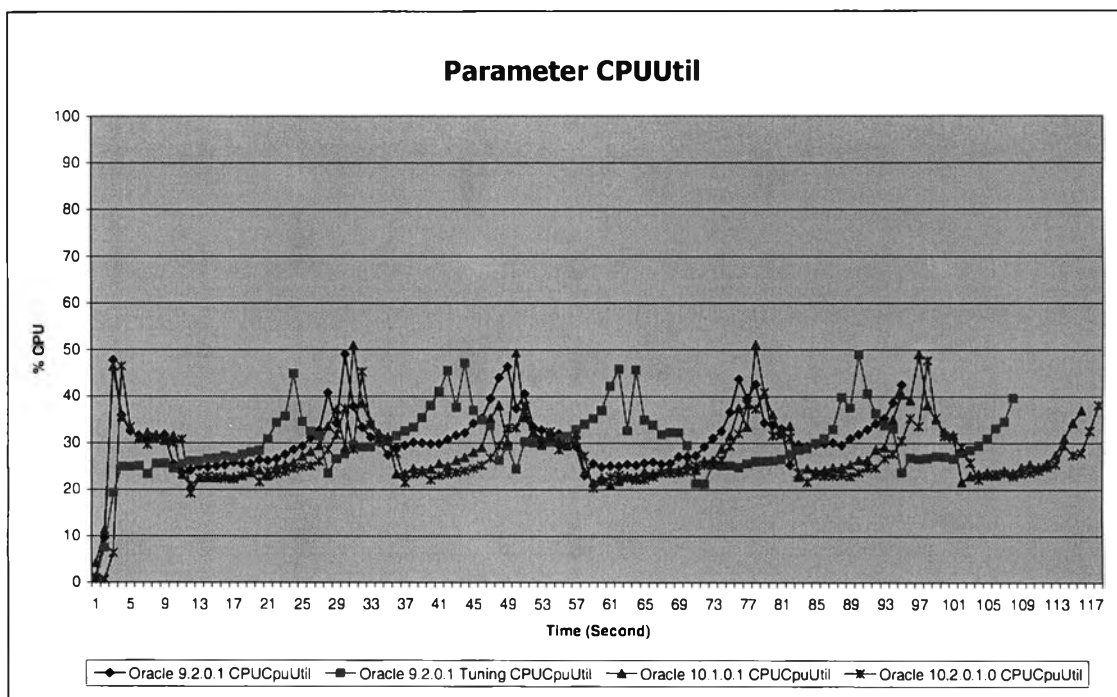


รูปที่ ค.42 แสดงการเปรียบเทียบพารามิเตอร์ MEMFreeMem ในการ insert partition

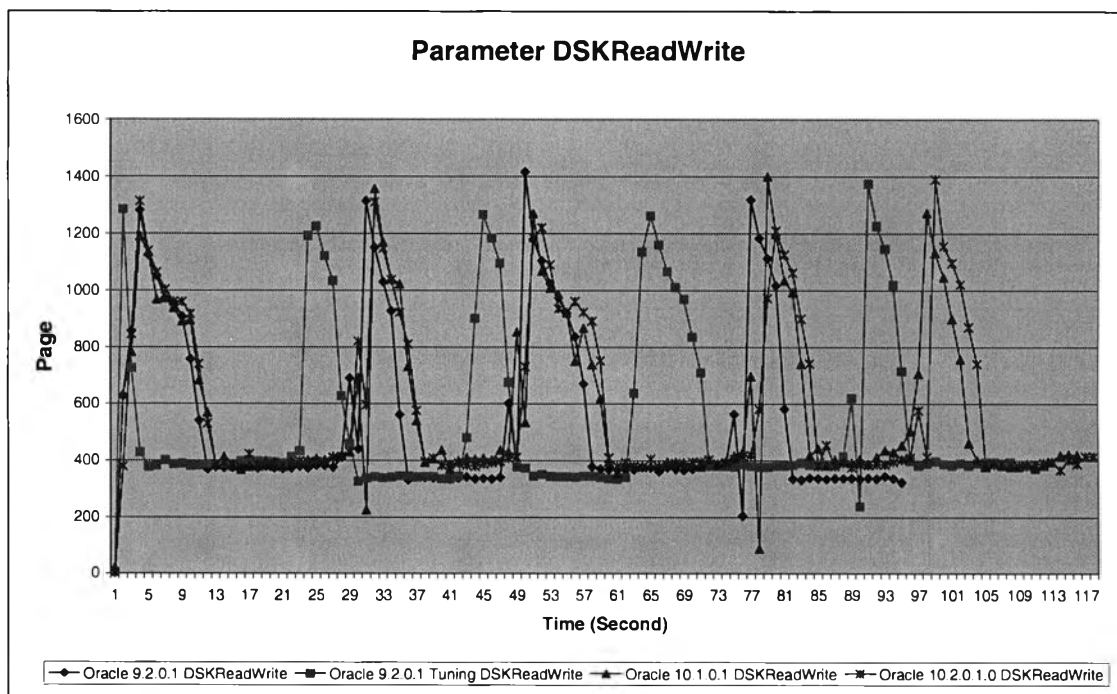
9) การจัดเก็บข้อมูลประจำวัน เป็นการนำข้อมูลที่เป็นเท็กซ็อนำเข้าสู่ตารางโดยใช้โปรแกรมเอสคิวแอลโหลดเดอร์ (SQL Loader) เนื่องจากเท็กซ็ไฟล์ที่นำเข้านี้มีทั้งหมด 5 ไฟล์ ซึ่งเป็นไฟล์ที่มี การบีบอัด (Zip file) การทำงานจึงประกอบด้วย การก๊อปปี้ การแตกการบีบอัด และใช้เอสคิวแอลโหลดเดอร์ นำข้อมูลเข้าสู่ตาราง



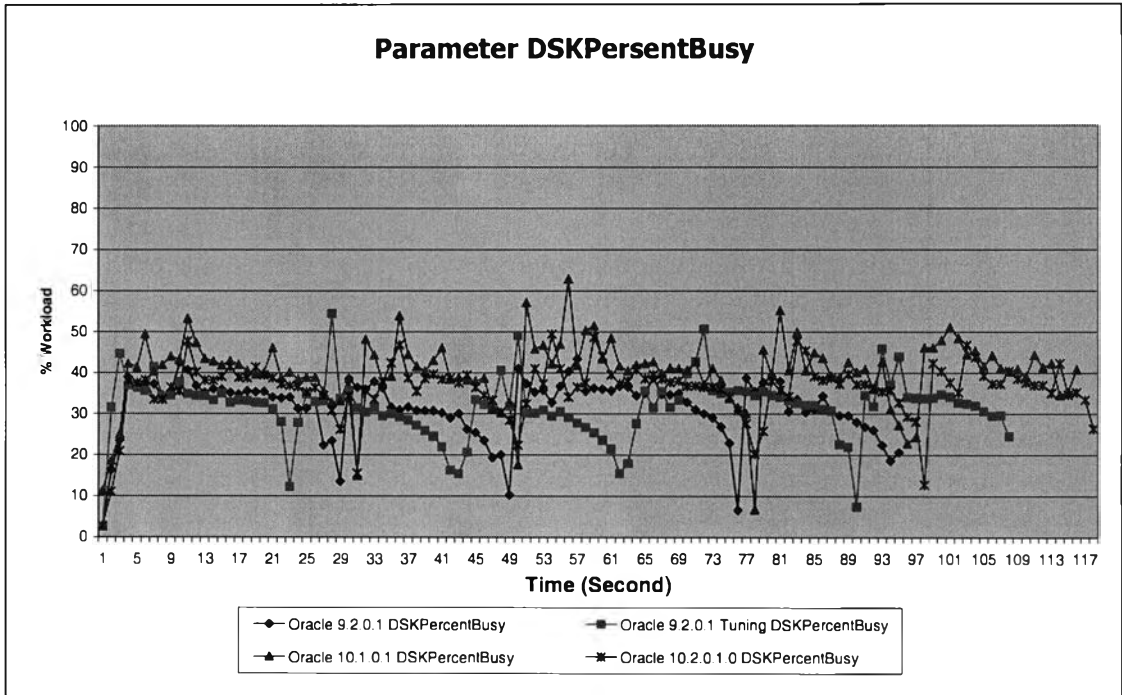
รูปที่ ค.43 แสดงการเปรียบเทียบพารามิเตอร์ CPUWio และ CPUIdleTime ในการใช้ sqldr



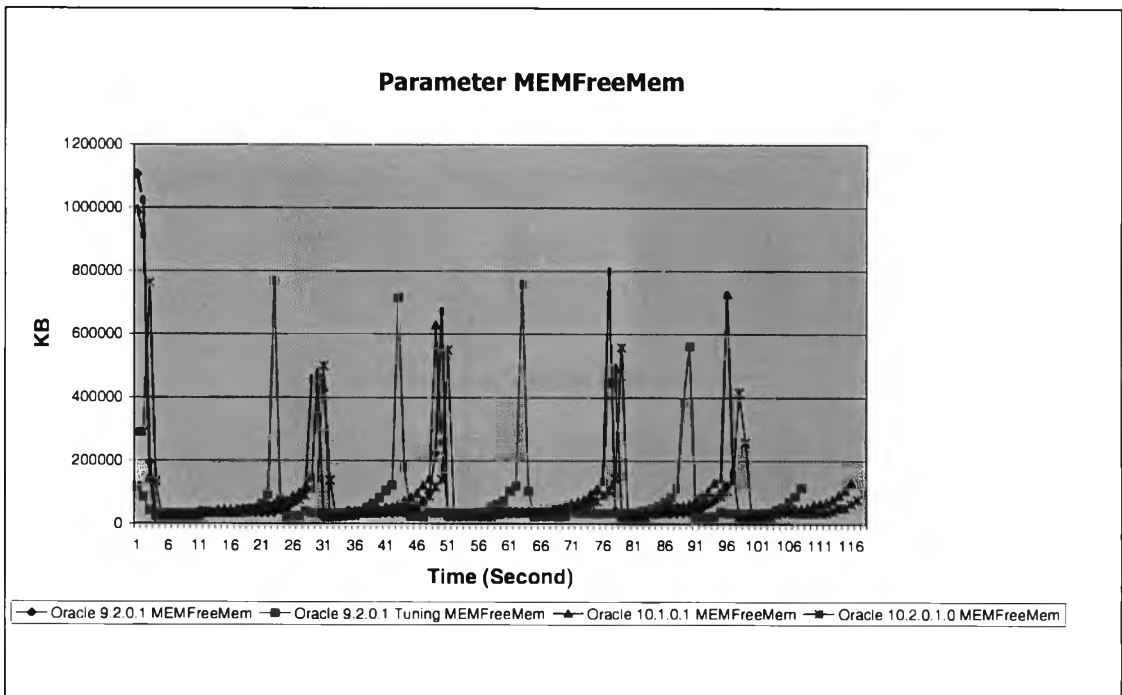
รูปที่ ค.44 แสดงการเปรียบเทียบพารามิเตอร์ CPUUtil ในการใช้ sqlldr



รูปที่ ค.45 แสดงการเปรียบเทียบพารามิเตอร์ DSKReadWrite ในการใช้ sqlldr

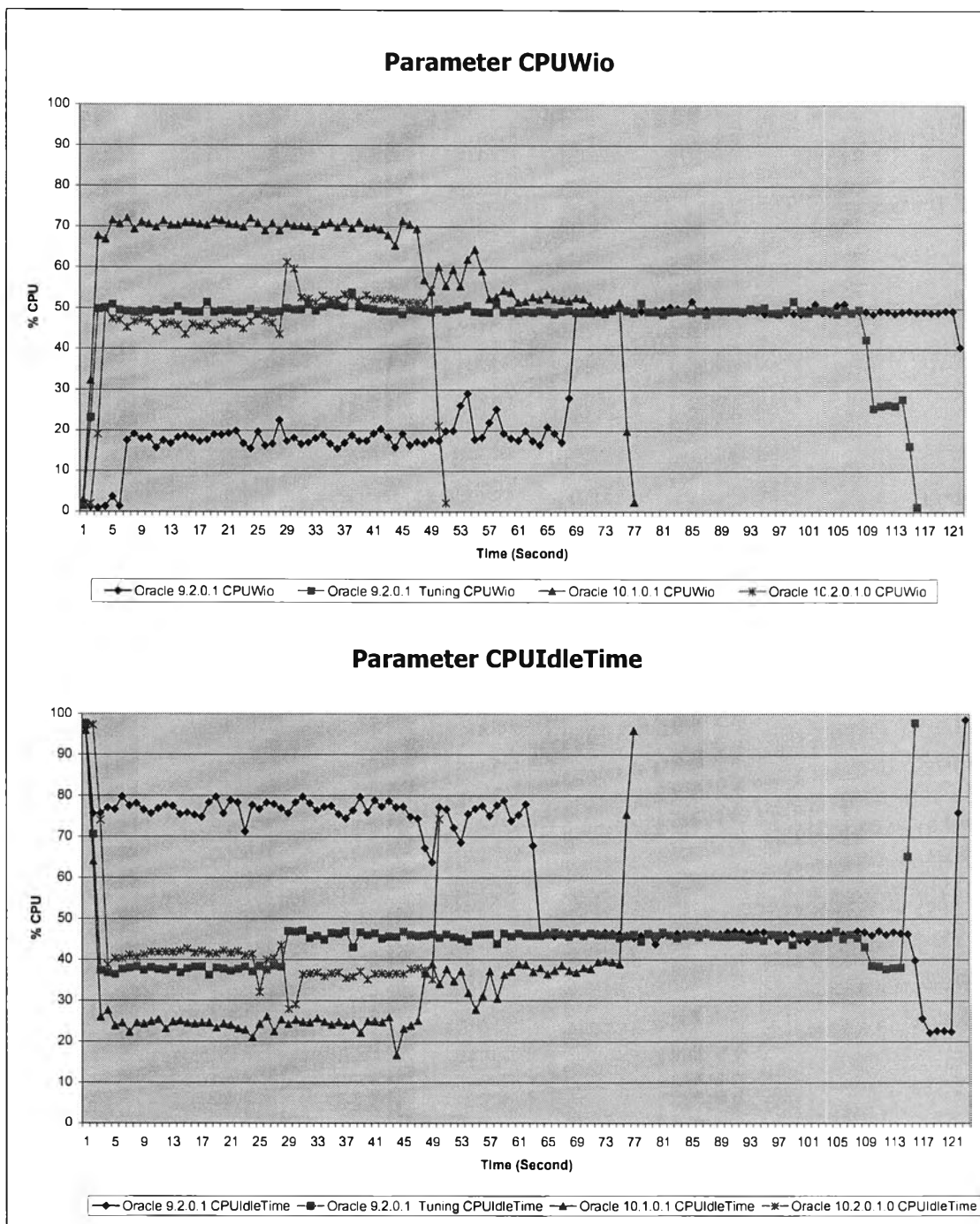


รูปที่ ค.46 แสดงการเปรียบเทียบพารามิเตอร์ DSKPresentBusy ในการใช้ sqldr

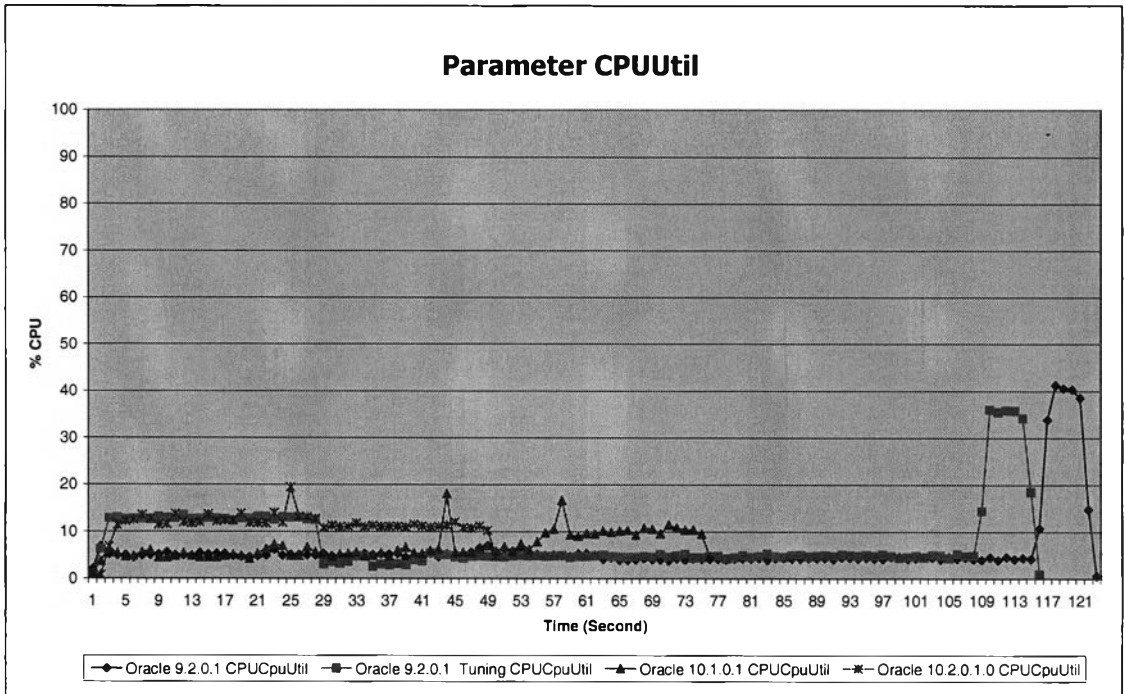


รูปที่ ค.47 แสดงการเปรียบเทียบพารามิเตอร์ DSKMEMFreeMem ในการใช้ sqldr

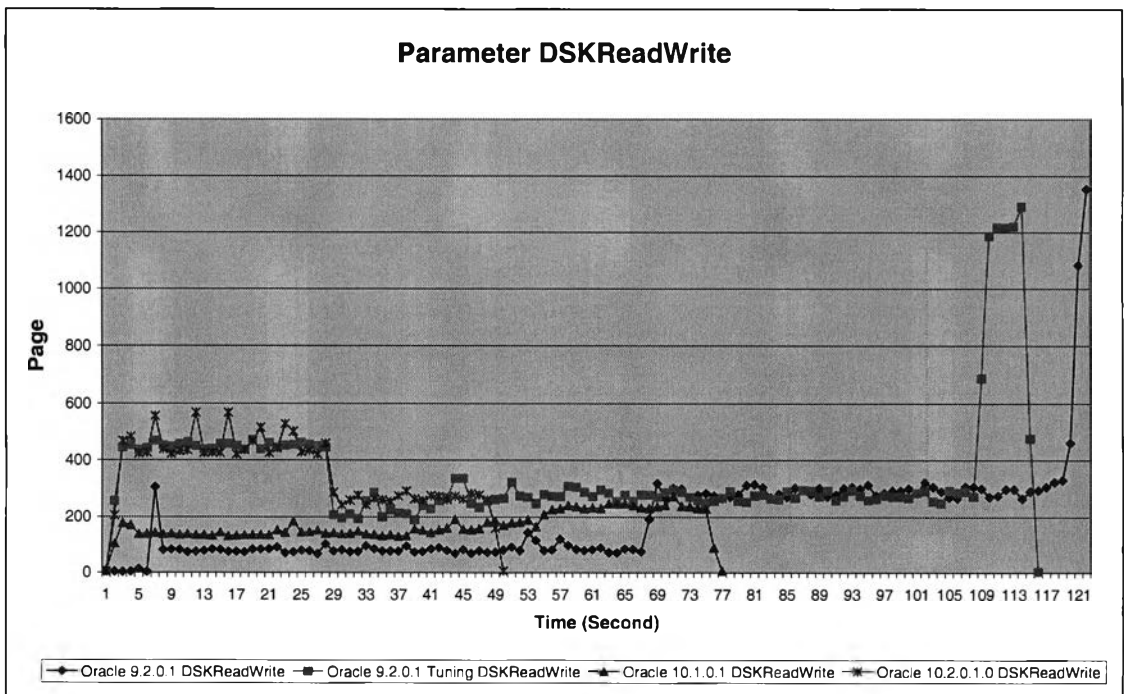
10) การปรับปรุงพาดิชั่นอินเด็กซ์ ในการใช้ทรัพยากรนั้นจะเห็นได้ว่า การจัดการฐานข้อมูลอราเคิลเวอร์ชัน 10.1.0.2 และ อราเคิลเวอร์ชัน 10.2.0.1 นั้นมีการจัดการที่ดีกว่า อราเคิลเวอร์ชัน 9.2.0.1



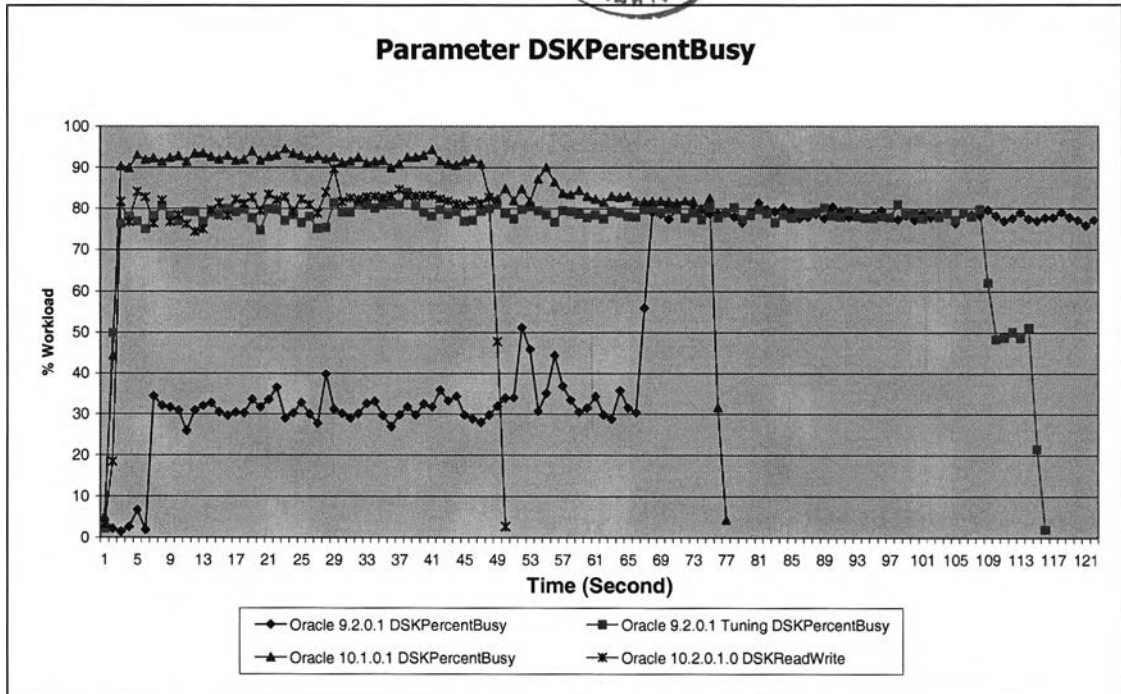
รูปที่ ค.48 แสดงการเปรียบเทียบพารามิเตอร์ CPUWio และ CPUIdleTime ในการ Rebuild partition index



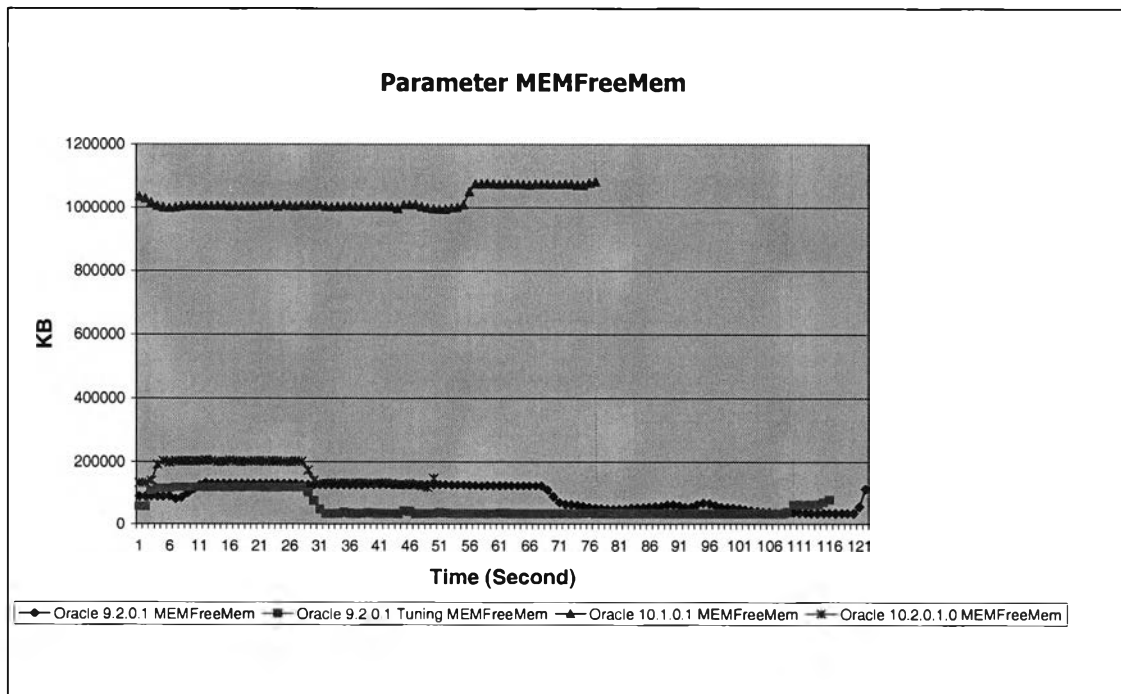
รูปที่ ค.49 แสดงการเปรียบเทียบพารามิเตอร์ CPUUtil ในการ Rebuild partition index



รูปที่ ค.50 แสดงการเปรียบเทียบพารามิเตอร์ DSKReadWrite ในการ Rebuild partition index

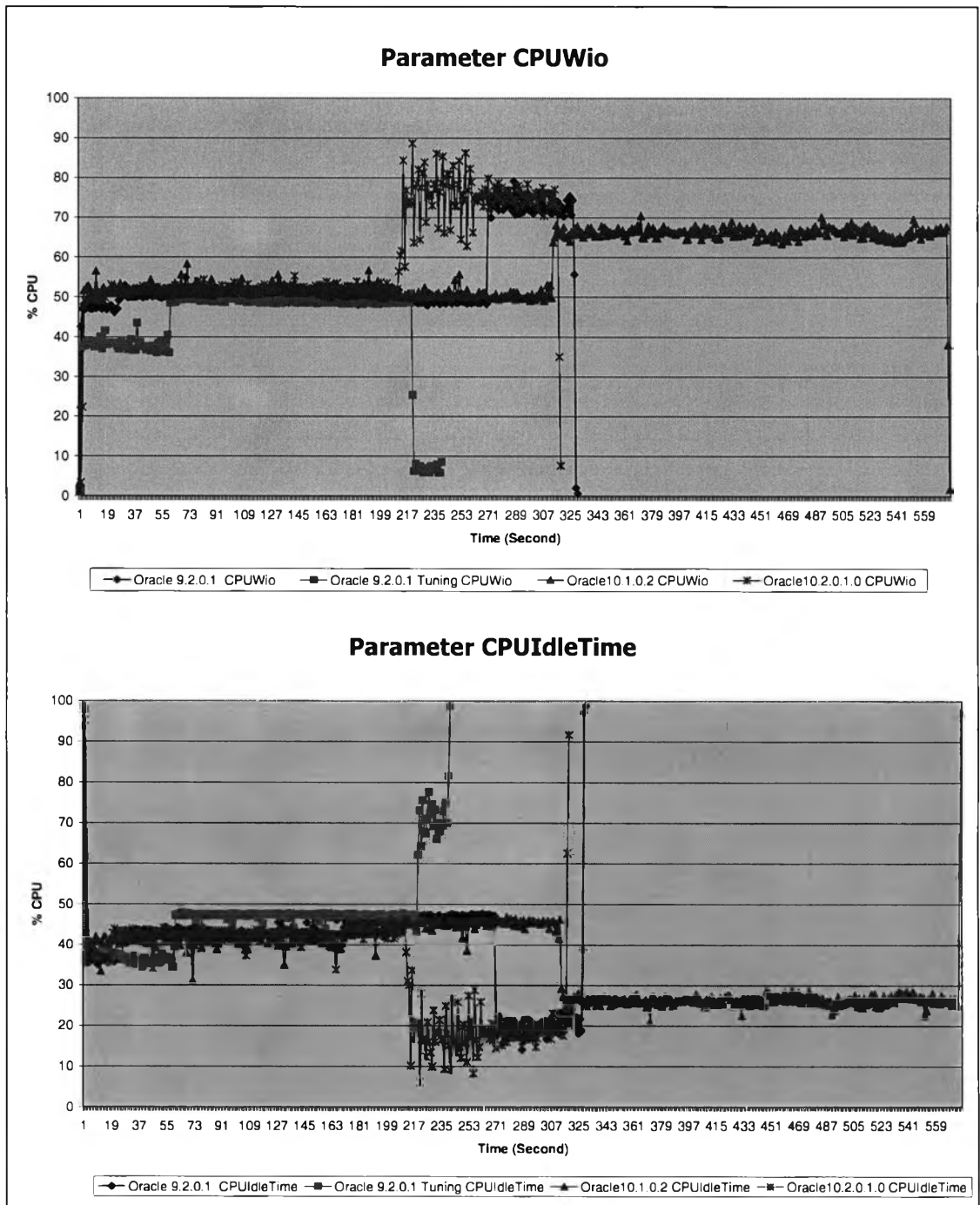


รูปที่ ค.51 แสดงการเปรียบเทียบพารามิเตอร์ DSKPercentBusy ในการ Rebuild partition index

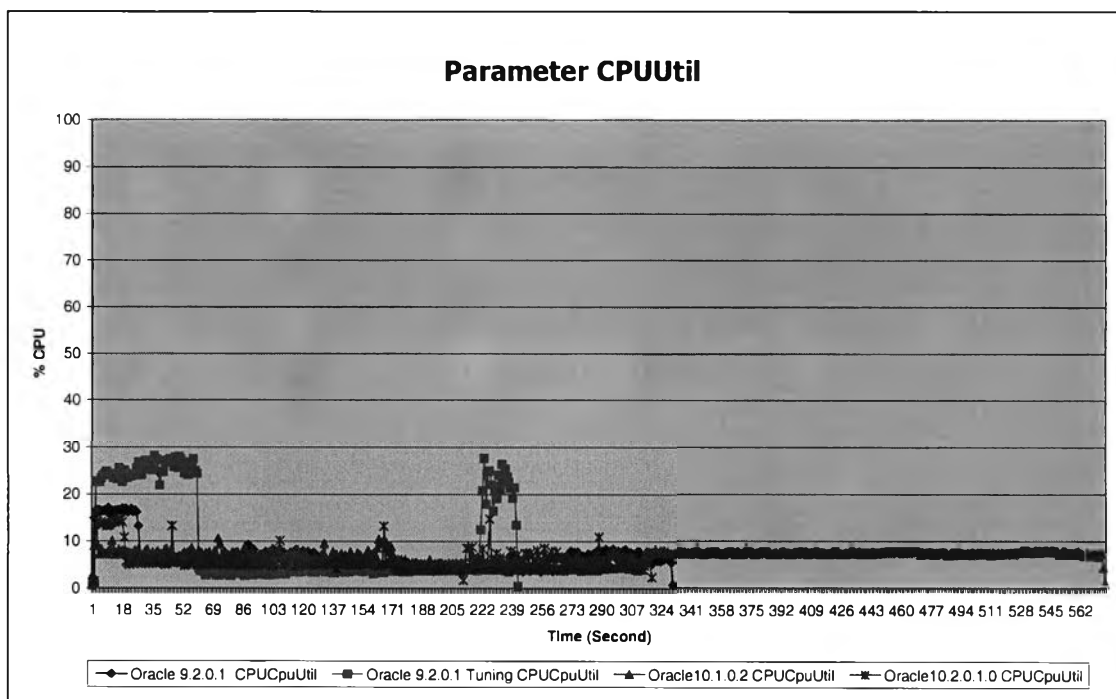


รูปที่ ค.52 แสดงการเปรียบเทียบพารามิเตอร์ MEMFreeMem ในการ Rebuild partition index

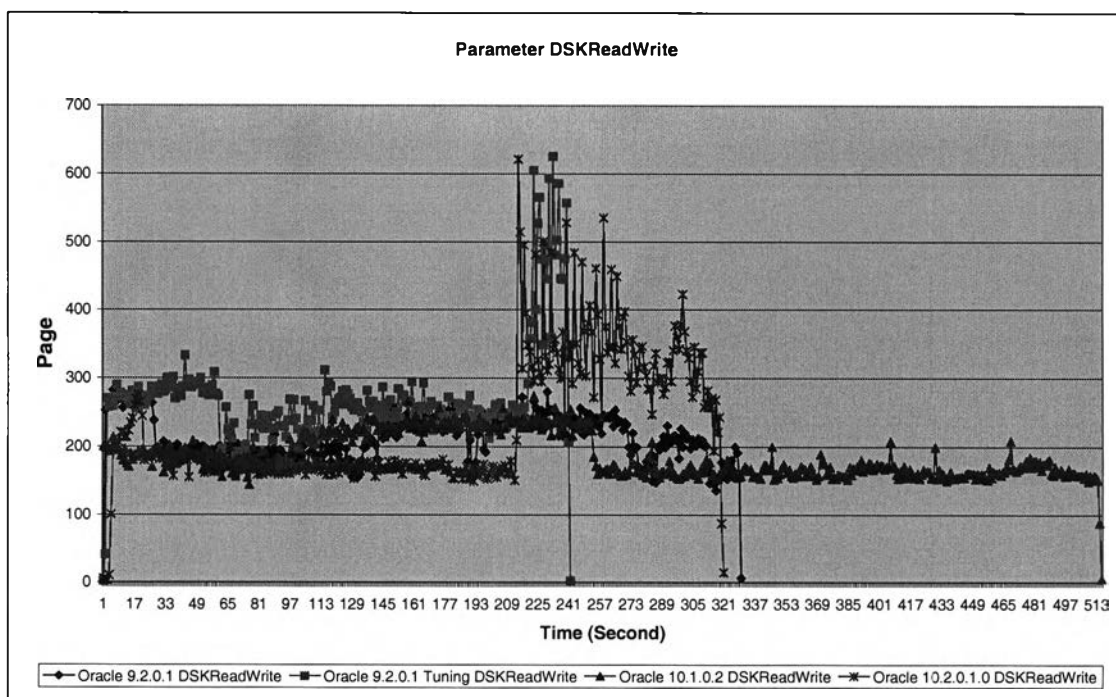
11) การใช้ฟังก์ชันสรุปยอด summary ออราเคิลเวอร์ชัน 10.1.0.2 มีการบริหารทรัพยากร
 ไม่ได้เท่าที่ควร



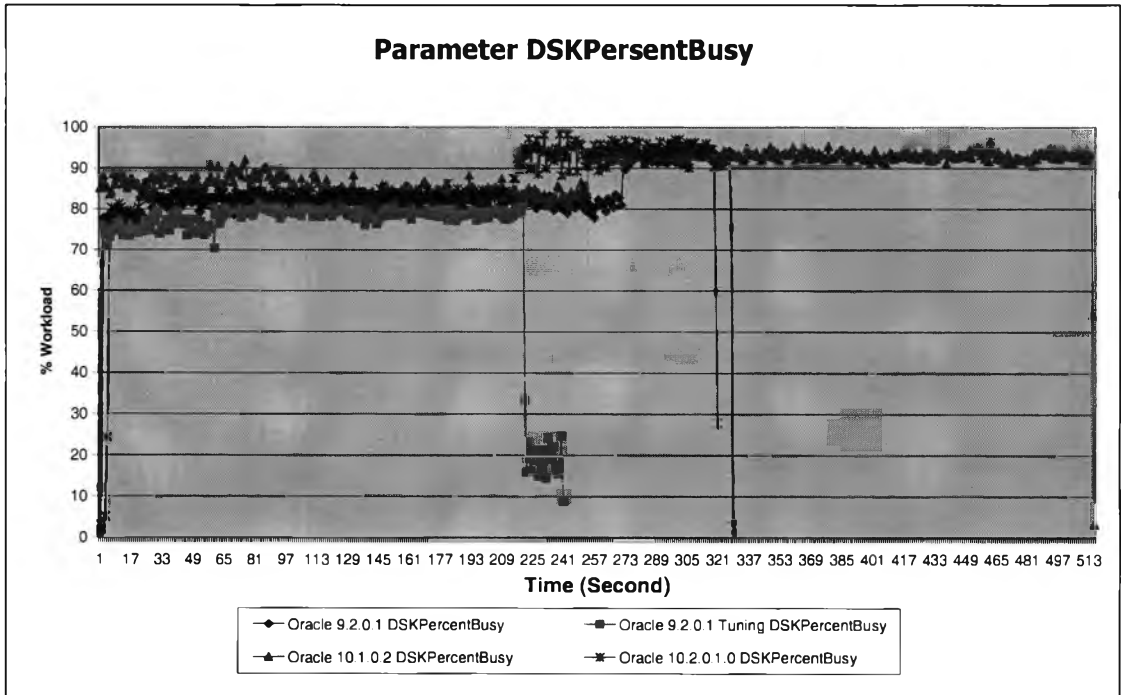
รูปที่ ค.53 แสดงการเปรียบเทียบพารามิเตอร์ CPUWio และ CPUIdleTime ในการทำ summary



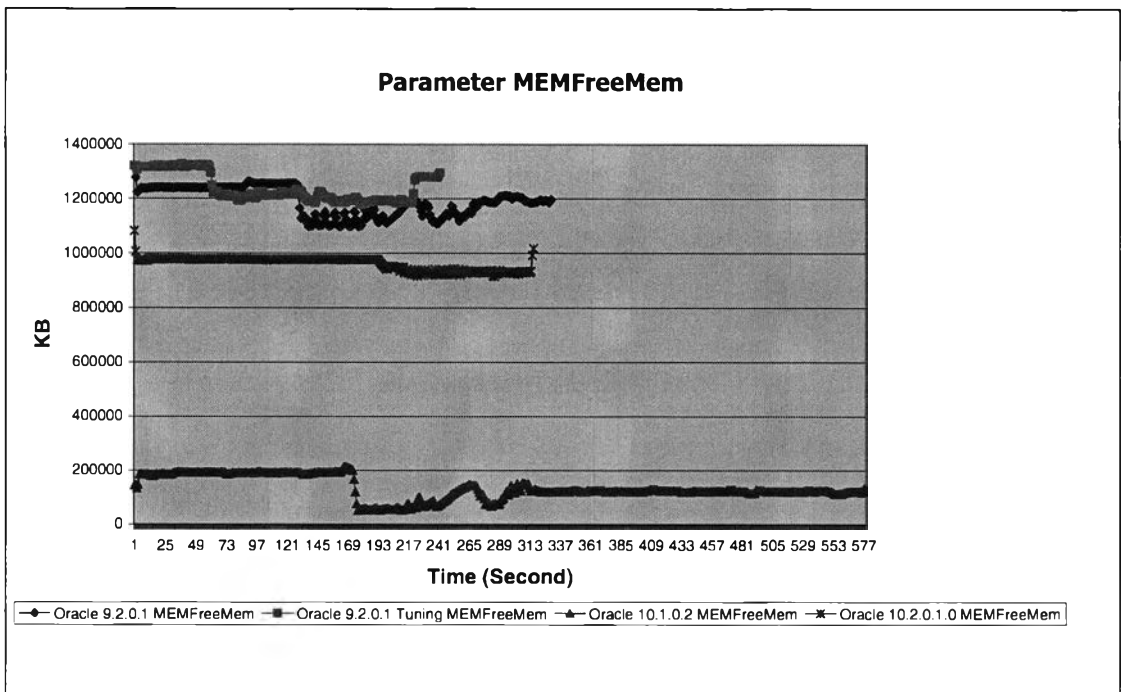
รูปที่ ค.54 แสดงการเปรียบเทียบพารามิเตอร์ CPUUtil ในการทำ summary



รูปที่ ค.55 แสดงการเปรียบเทียบพารามิเตอร์ DSKReadWrite ในการทำ summary

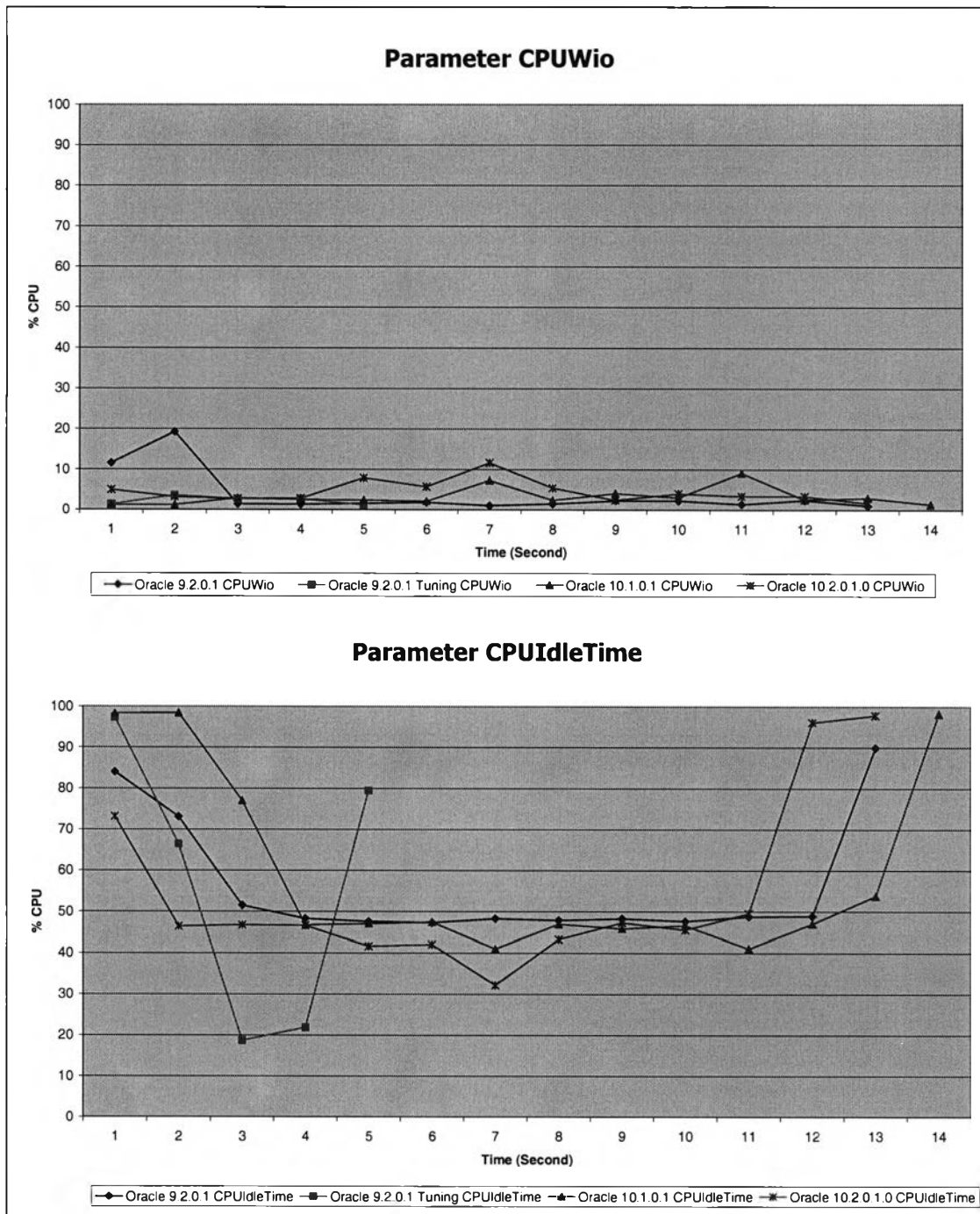


รูปที่ ค.56 แสดงการเปรียบเทียบพารามิเตอร์ DSKPercentBusy ในการทำ summary

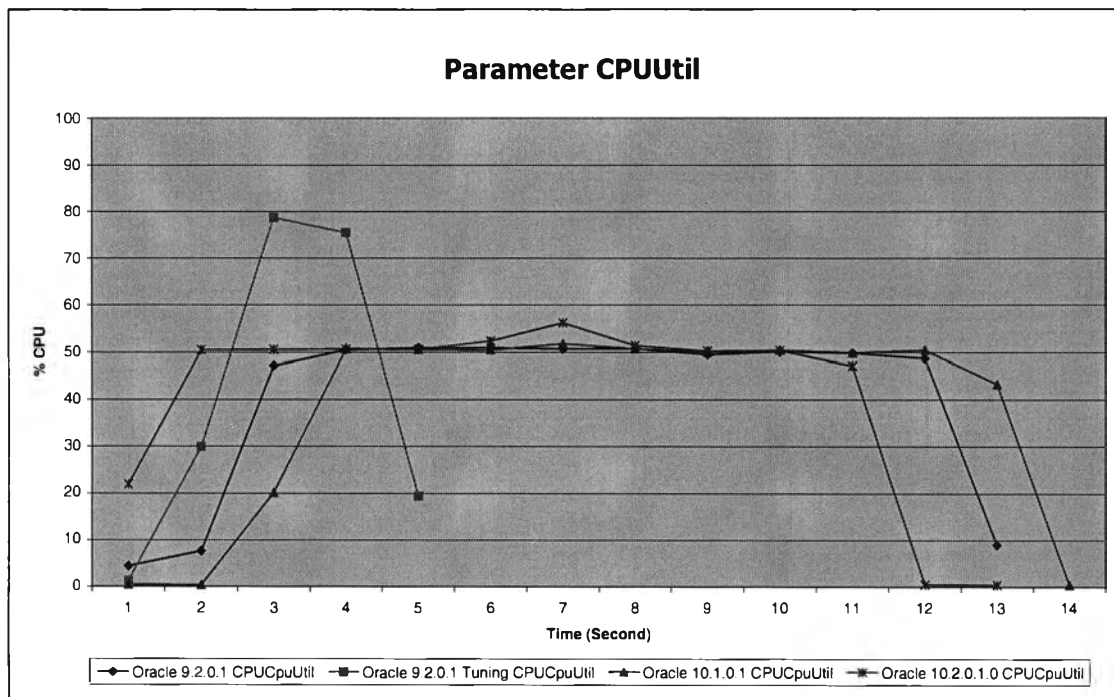


รูปที่ ค.57 แสดงการเปรียบเทียบพารามิเตอร์ MEMFreeMem ในการทำ summary

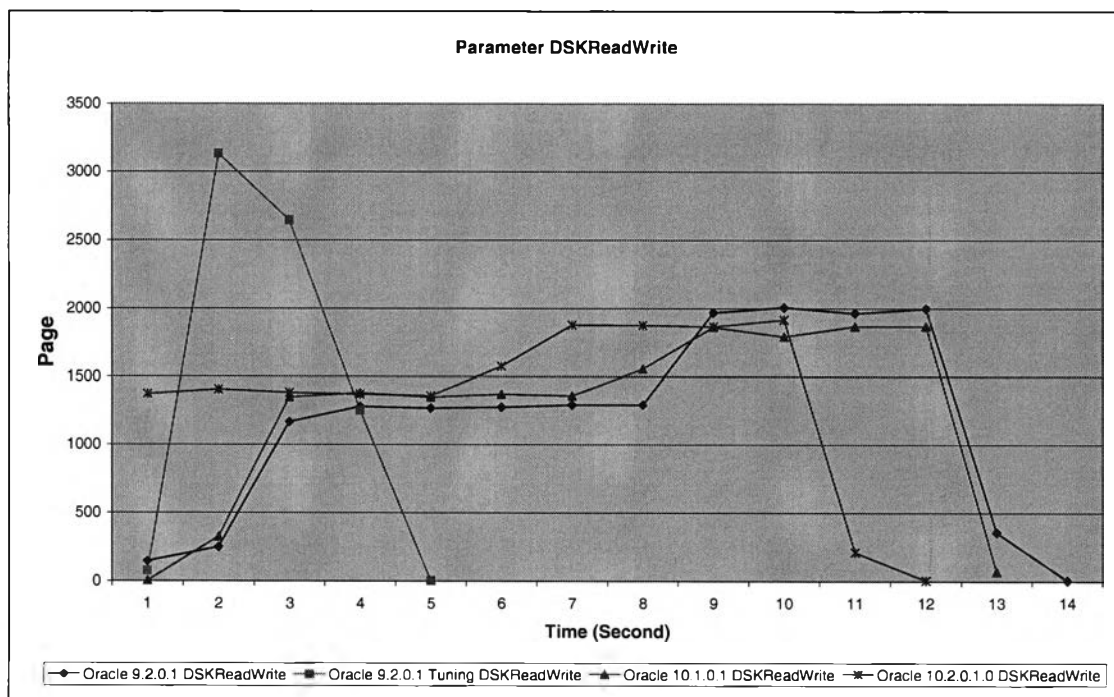
12) การจัดเรียงข้อมูล โดยการใช้คำสั่ง *Group by* และ *Order by*



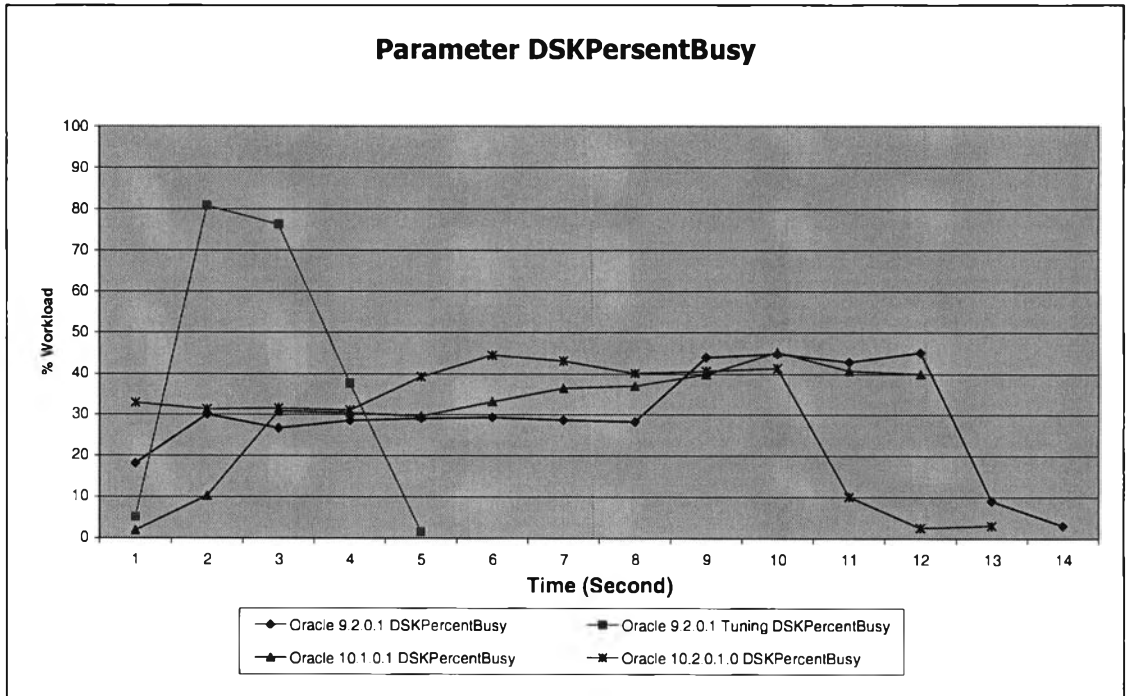
รูปที่ ค.58 แสดงการเปรียบเทียบพารามิเตอร์ CPUWio และ CPUIdleTime ในการทำ sorting



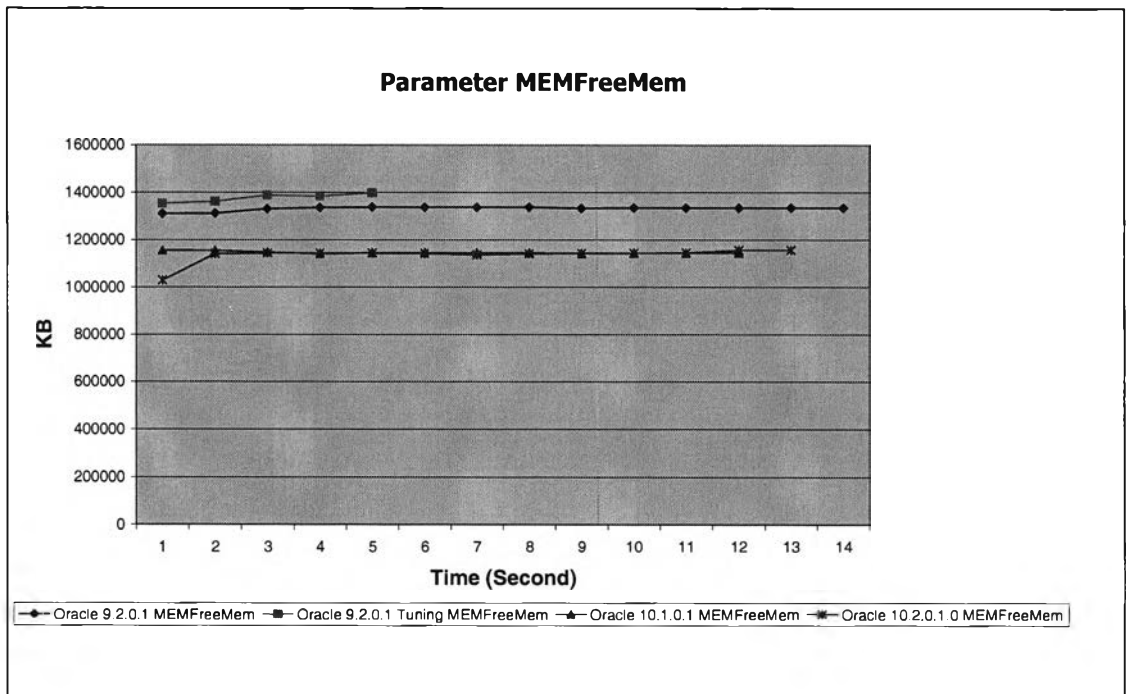
รูปที่ ค.59 แสดงการเปรียบเทียบพารามิเตอร์ CPUUtil ในการทำ sorting



รูปที่ ค.60 แสดงการเปรียบเทียบพารามิเตอร์ DSKReadWrite ในการทำ sorting

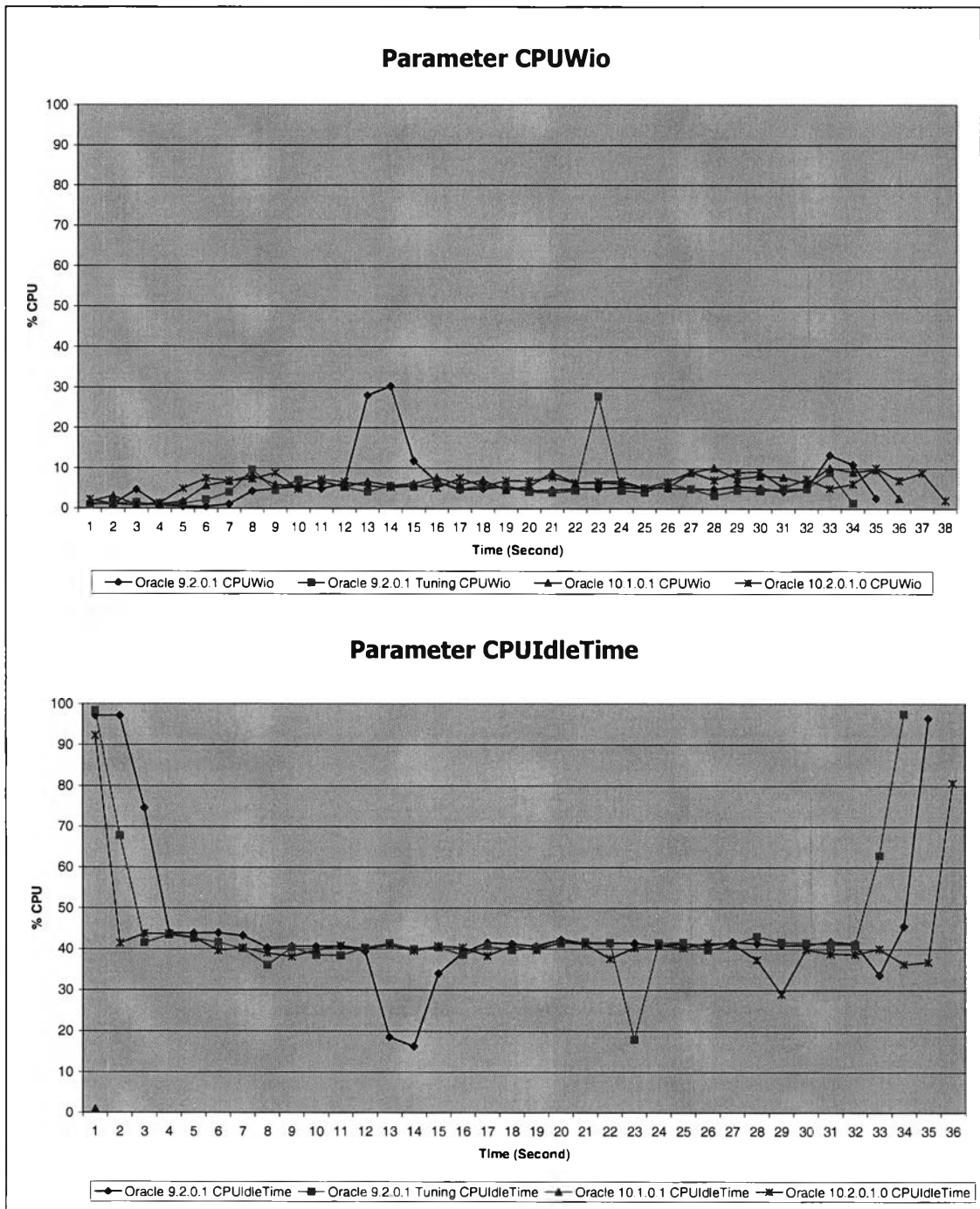


รูปที่ ค.61 แสดงการเปรียบเทียบพารามิเตอร์ DSKPersetBusy ในการทำ sorting

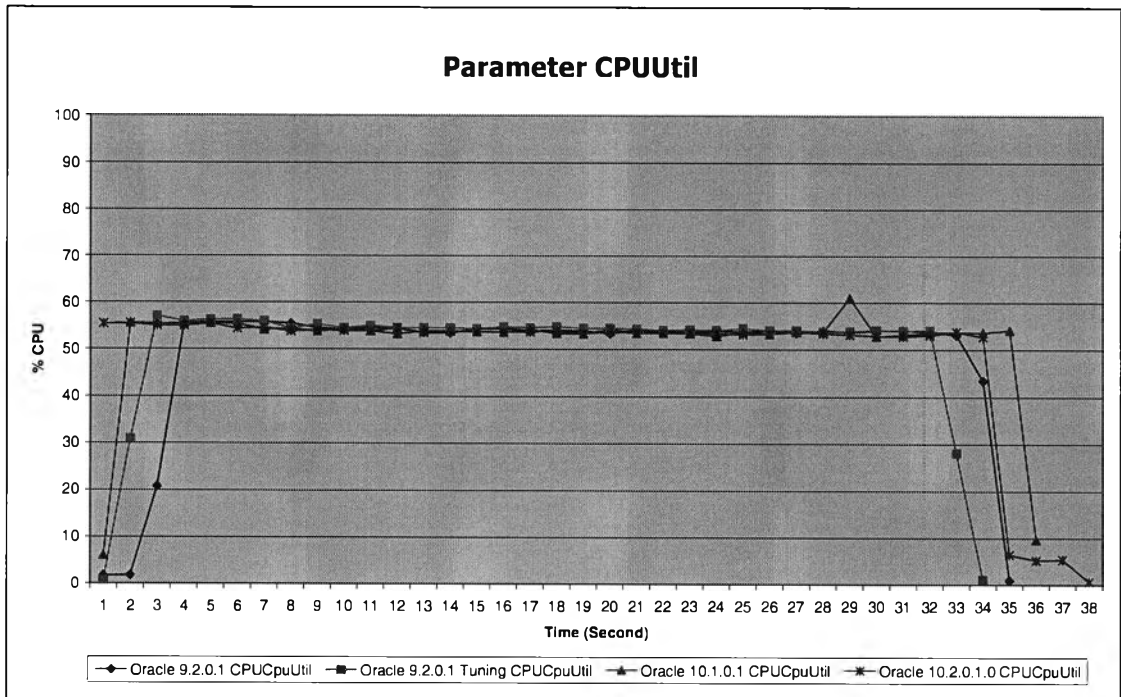


รูปที่ ค.62 แสดงการเปรียบเทียบพารามิเตอร์ MEMFreeMem ในการทำ sorting

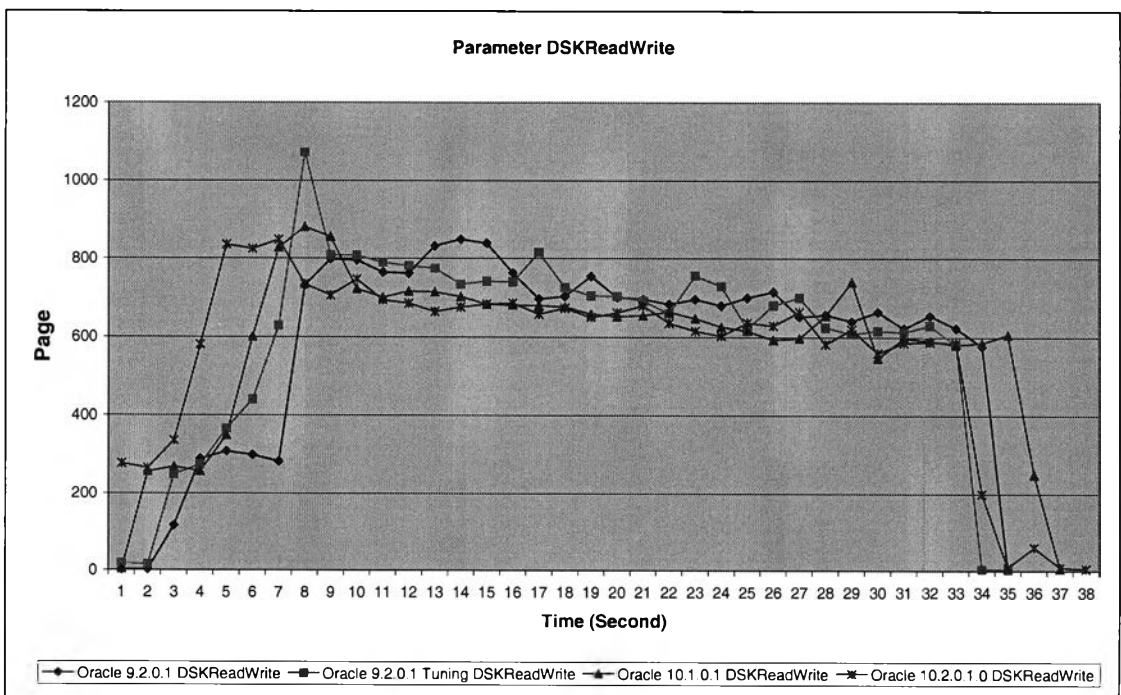
13) การนำข้อมูลออกจากตารางโดยการเอ็กซ์พอร์ต จะเห็นได้ว่าในแต่ละเวอร์ชันการทำงานนั้นไม่มีความแตกต่างกัน



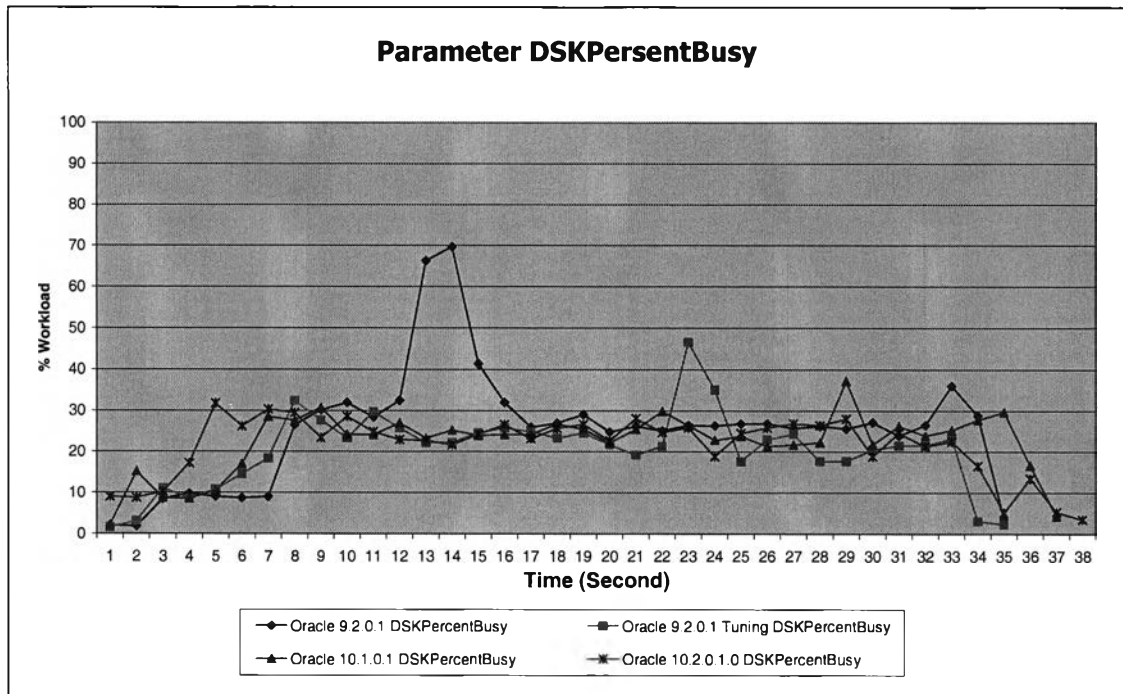
รูปที่ ค.63 แสดงการเปรียบเทียบพารามิเตอร์ CPUWio และ CPUIdleTime ในการ export



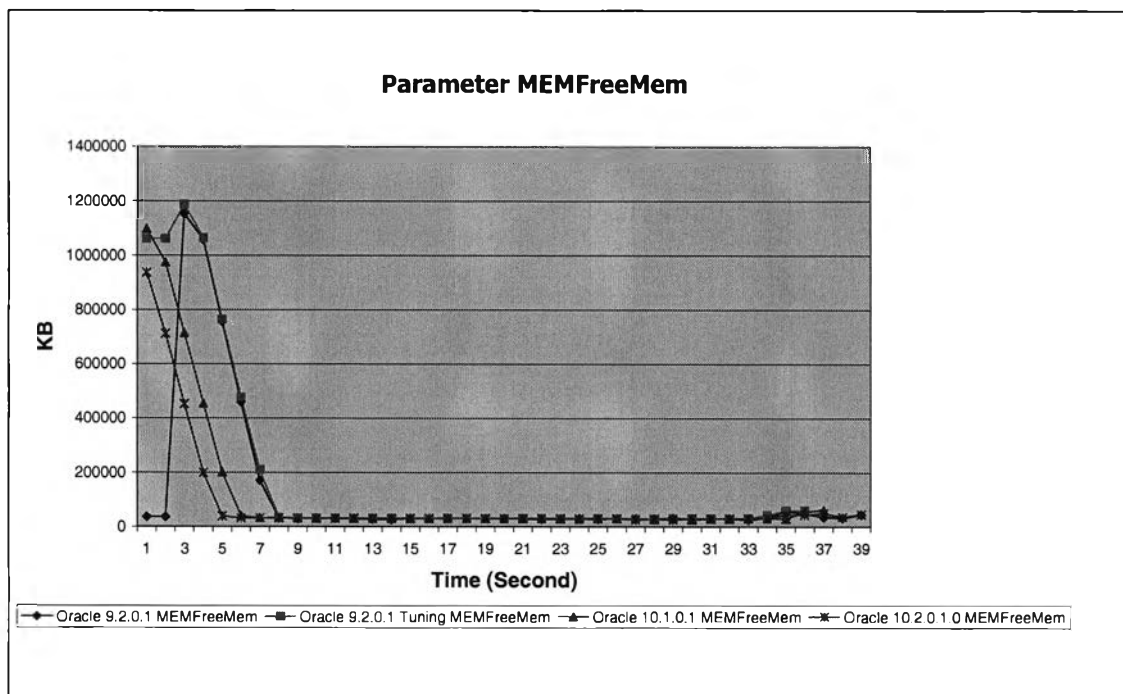
รูปที่ ค.64 แสดงการเปรียบเทียบพารามิเตอร์ CPUUtil ในการ export



รูปที่ ค.65 แสดงการเปรียบเทียบพารามิเตอร์ DSKRead/Write ในการ export

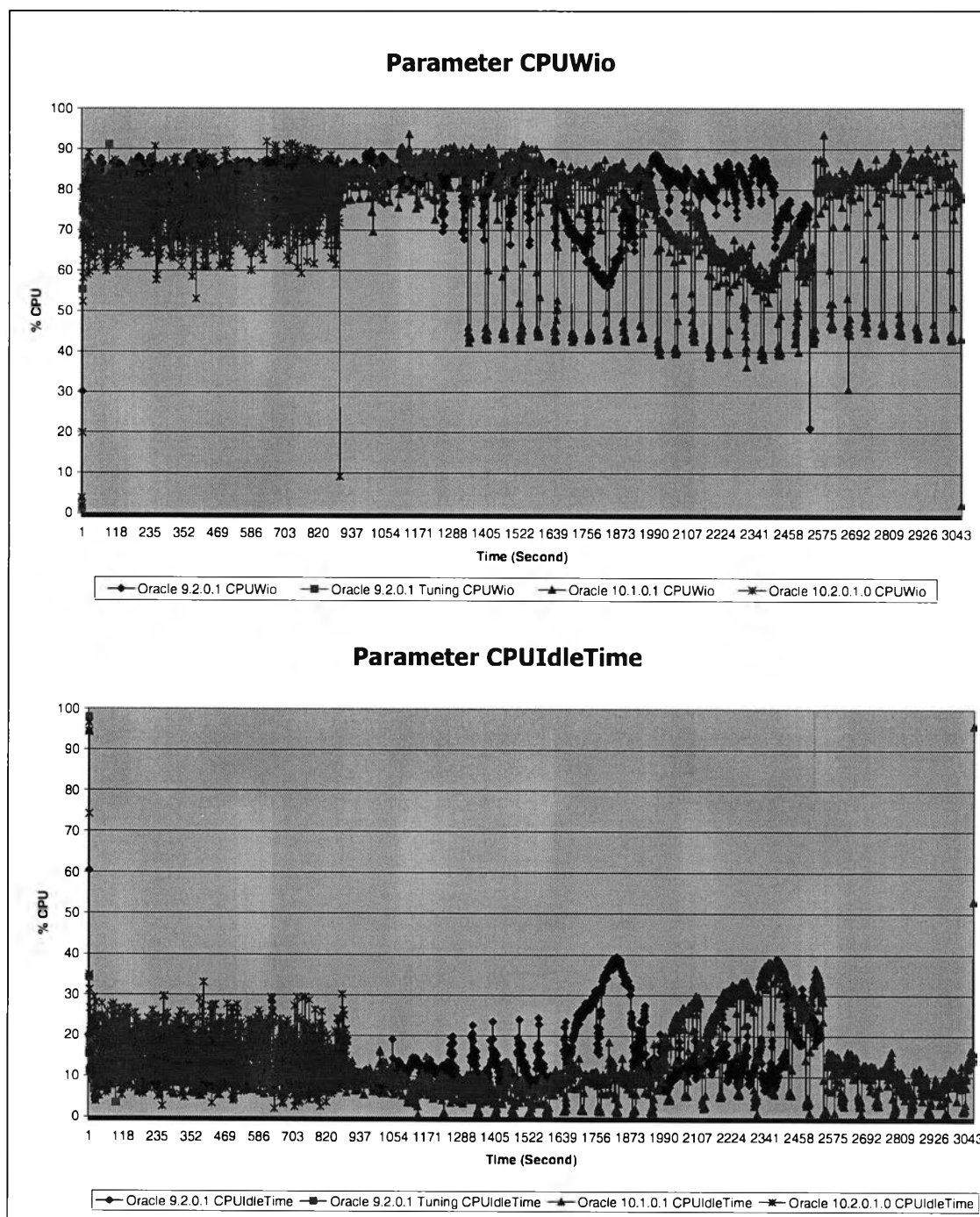


รูปที่ ค.66 แสดงการเปรียบเทียบพารามิเตอร์ DSKPercentBusy ในการ export

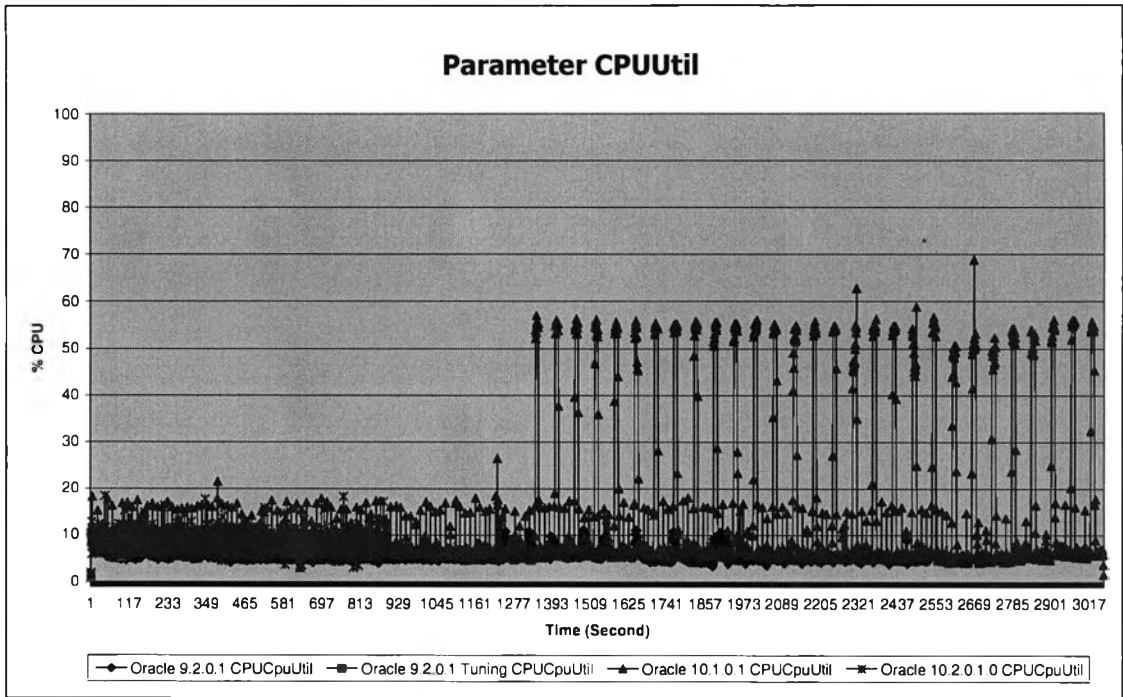


รูปที่ ค.67 แสดงการเปรียบเทียบพารามิเตอร์ MEMFreeMem ในการ export

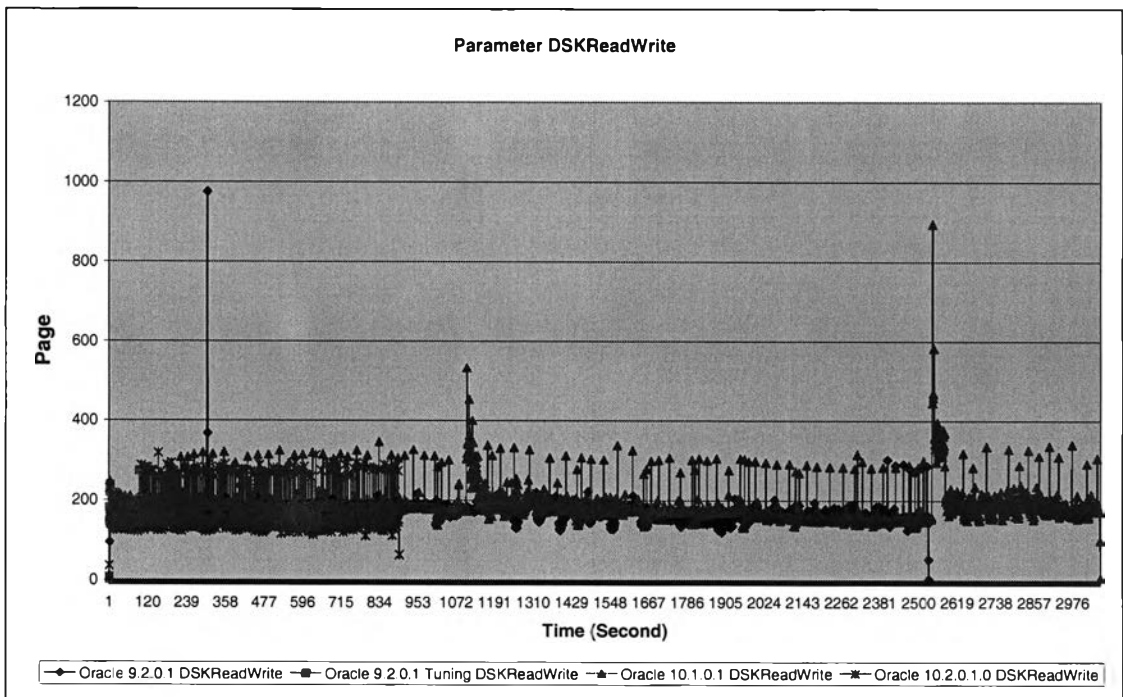
14) การนำข้อมูลเข้าตารางโดยการอิมพอร์ต ในการทำงานนี้ใน การเพิ่มประสิทธิภาพนั้น จะไม่มีการใช้อินเด็กซ์ในตารางการทำงาน ทำให้มีการทำงานเร็วขึ้นมาก



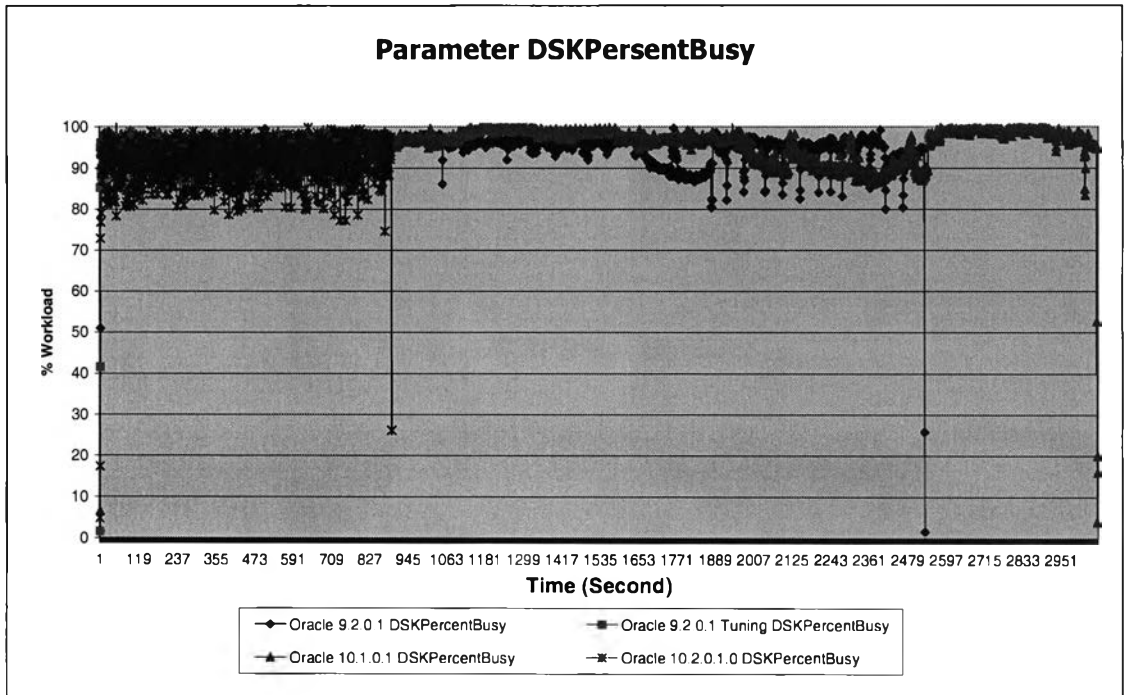
รูปที่ ค.68 แสดงการเปรียบเทียบพารามิเตอร์ CPUWio และ CPUIdleTime ในการ import



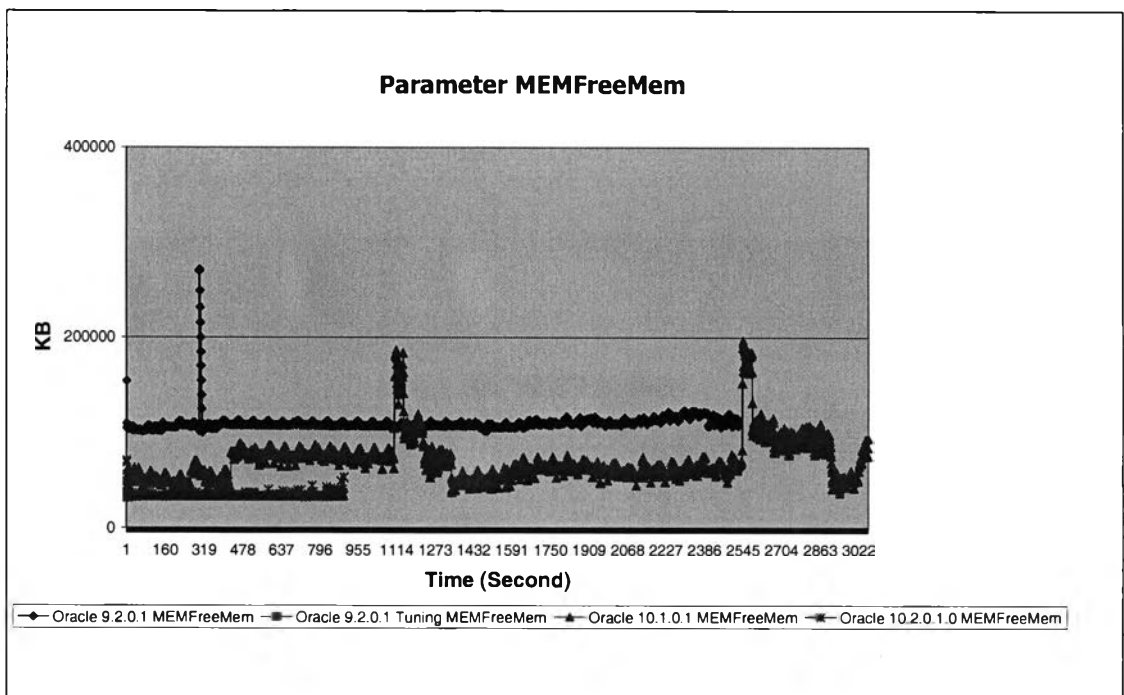
รูปที่ ค.69 แสดงการเปรียบเทียบพารามิเตอร์ CPUUtil ในการ import



รูปที่ ค.70 แสดงการเปรียบเทียบพารามิเตอร์ DSKReadWrite ในการ import

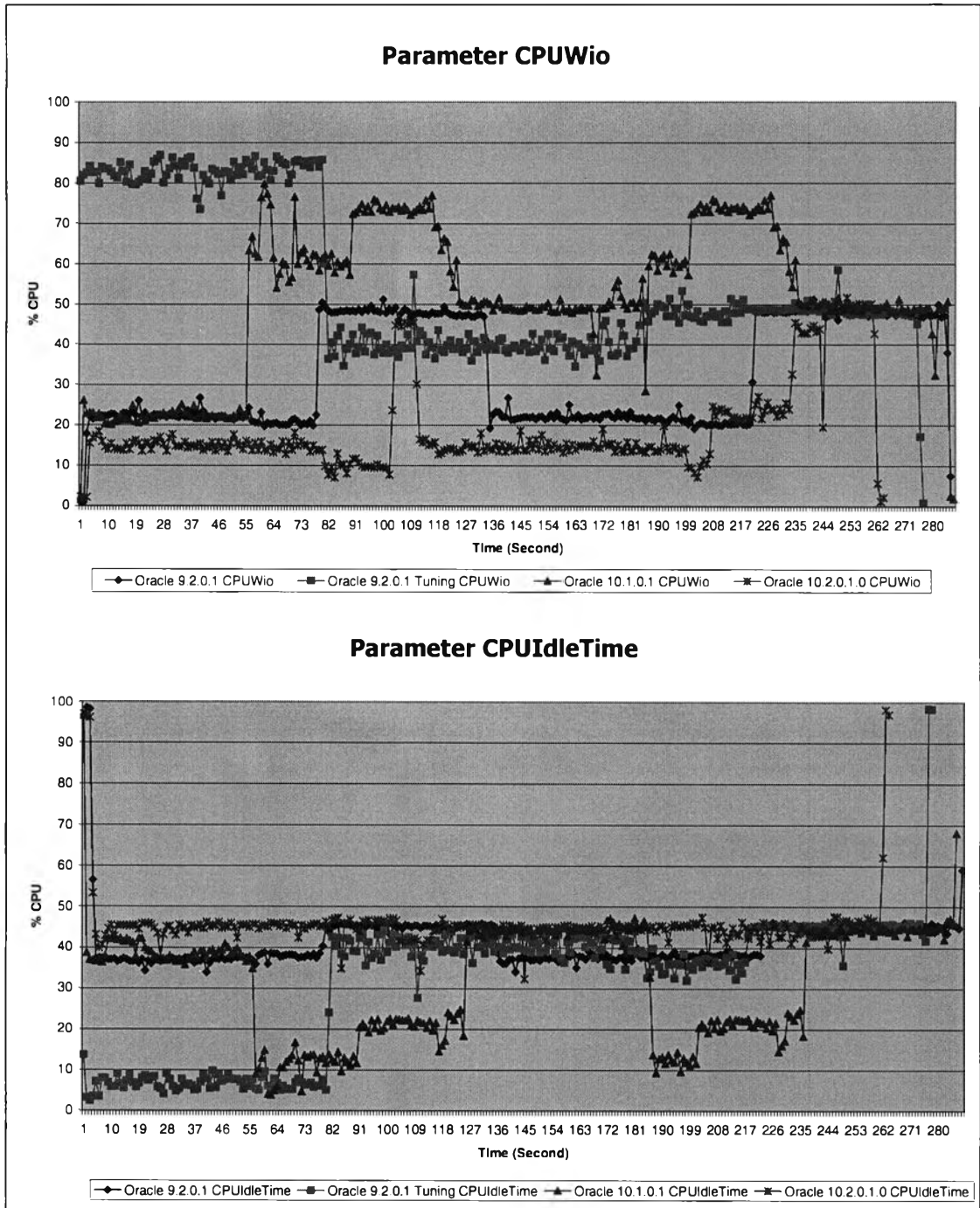


รูปที่ ค.71 แสดงการเปรียบเทียบพารามิเตอร์ DSKPersetBusy ในการ import

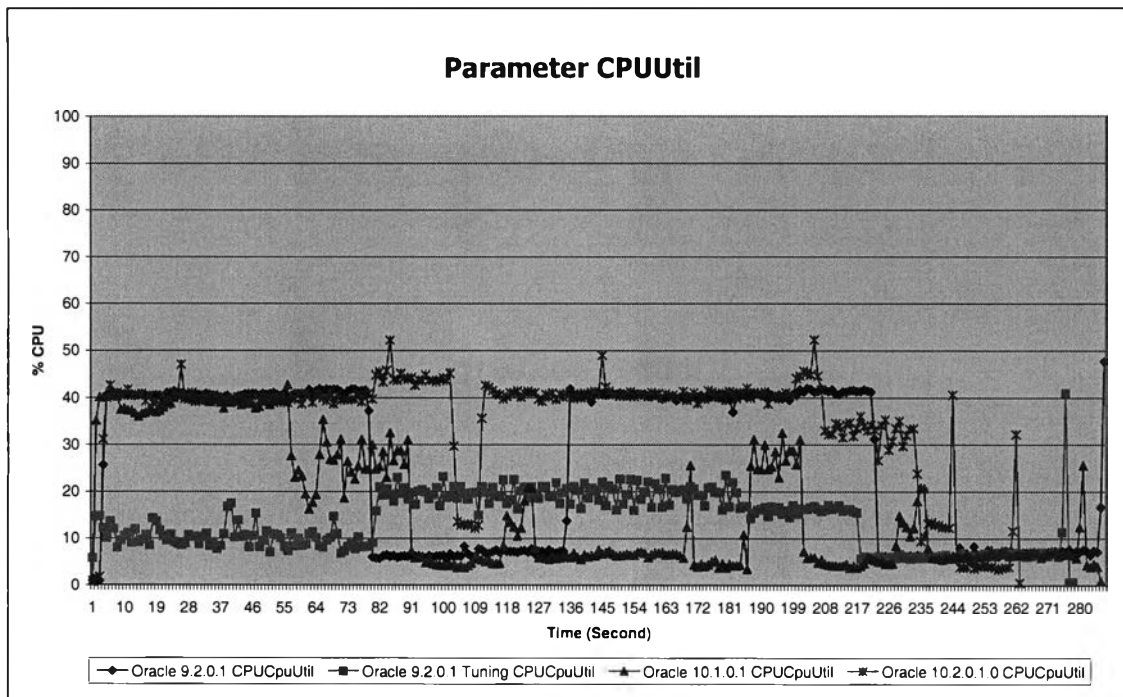


รูปที่ ค.72 แสดงการเปรียบเทียบพารามิเตอร์ MEMFreeMem ในการ import

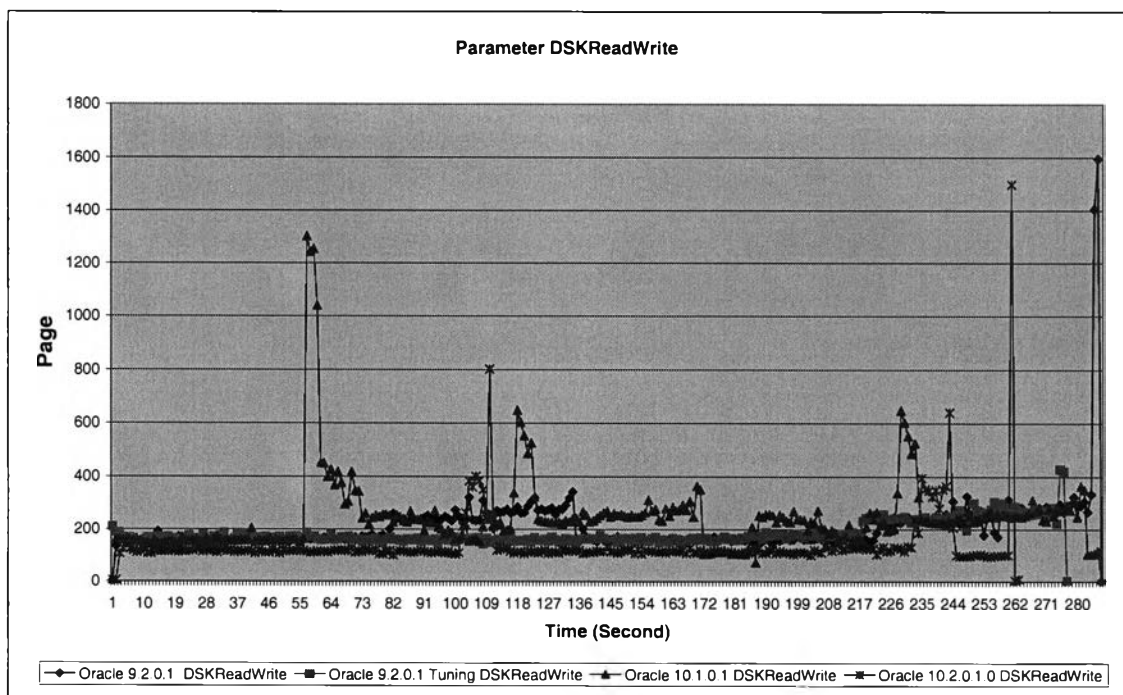
15) การทำวิเคราะห์ตาราง Analyze



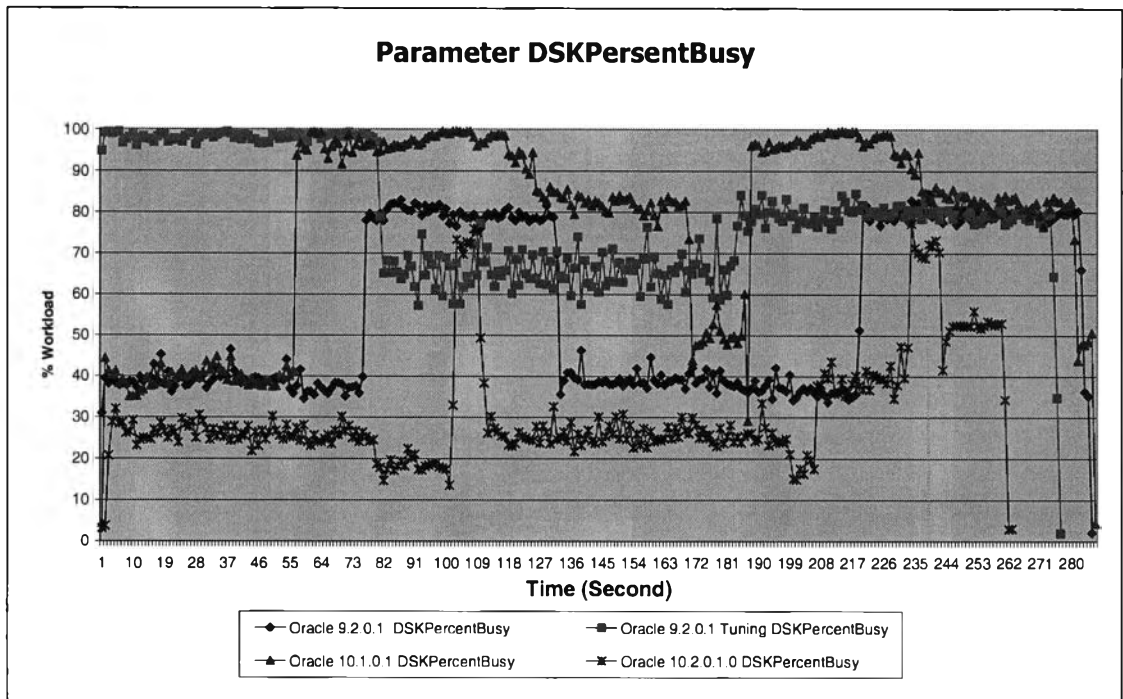
รูปที่ ค.73 แสดงการเปรียบเทียบพารามิเตอร์ CPUWio และ CPUIdleTime ในการทำ Analyze



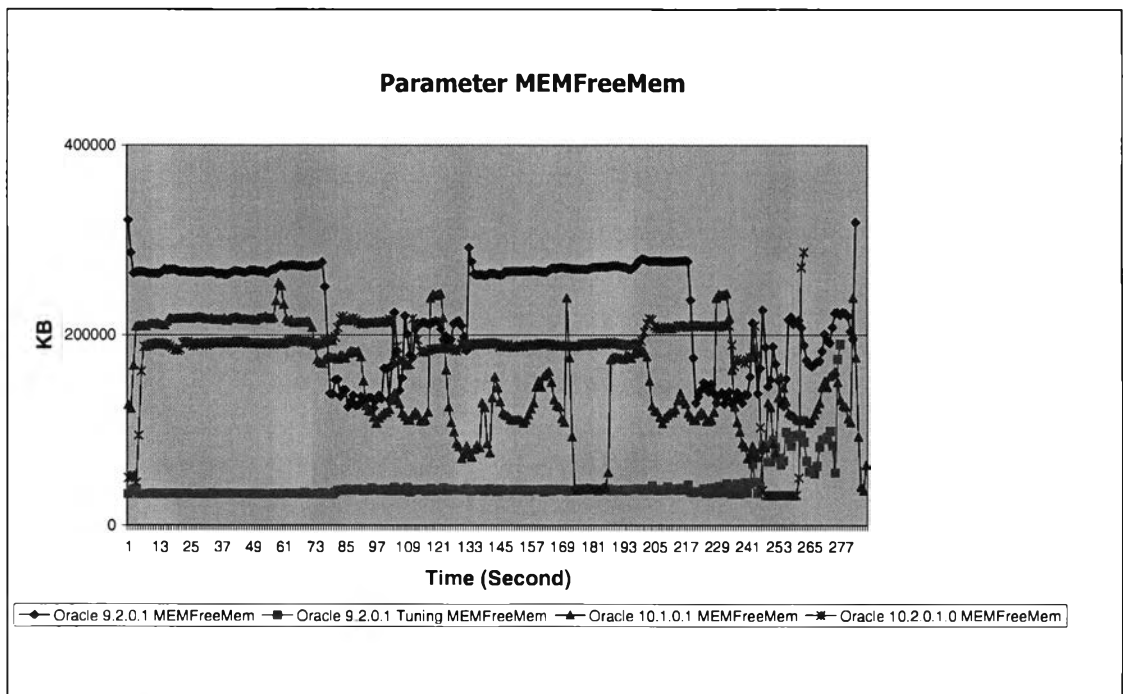
รูปที่ ค.74 แสดงการเปรียบเทียบพารามิเตอร์ CPUUtil ในการทำ Analyze



รูปที่ ค.75 แสดงการเปรียบเทียบพารามิเตอร์ DSKReadWrite ในการทำ Analyze



รูปที่ ค.76 แสดงการเปรียบเทียบพารามิเตอร์ DSKPercentBusy ในการทำ Analyze



รูปที่ ค.77 แสดงการเปรียบเทียบพารามิเตอร์ MEMFreeMem ในการทำ Analyze

ประวัติผู้เขียนวิทยานิพนธ์

นางสาวปาริชาติ แม่นสวัสดิ์ เกิดเมื่อวันที่ 9 พฤษภาคม 2515 จังหวัดพิษณุโลก ได้สำเร็จการศึกษาระดับปริญญาตรีวิทยาศาสตร์บัณฑิต สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยอุตรดิตถ์ ปีการศึกษา 2537 และเข้าศึกษาต่อในระดับปริญญาโท หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิทยาศาสตร์คอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ภาคปีการศึกษา 2548

