

## ภาคผนวก ก

### การติดตั้ง Oracle Replication

ก่อนที่จะเริ่มทำการติดตั้ง replication site ต้องทำการ set parameter ใน parameter file ดังนี้

- ต้องแน่ใจว่า GLOBAL\_NAMES เป็นค่า TRUE
- ต้องแน่ใจว่า job processes เพียงพอสำหรับจำนวน master site

#### Global Names

เพื่อให้แน่ใจว่าสิ่งแวดล้อมของการ Replication ทำงานได้อย่างเหมาะสม ต้องทำการ set parameter GLOBAL\_NAMES ให้มีค่าเป็น TRUE เพื่อให้ database link มีชื่อเดียวกับชื่อ database โดยทำการกำหนดค่าใน initialization parameter file ดังนี้

```
global_names = TRUE
```

#### Job Processes

การกำหนดค่า job process ที่เหมาะสมมีความสำคัญอย่างยิ่งในการรองรับสิ่งแวดล้อมของการ replication เนื่องจากมีส่วนในการถ่ายทอดการจัดการ queue, การเชื่อม transaction ของแต่ละ site และการ refresh snapshot และอื่นๆ

สำหรับ Multimaster replication นั้นแต่ละ site จะมี รายละเอียดการเชื่อมต่อ (scheduled link) ไปยังแต่ละ master site อื่น ตัวอย่างเช่นถ้า มี 6 master site ในระบบฐานข้อมูลแบบกระจาย แต่ละ site ก็ต้องมีรายละเอียดของการเชื่อมต่อไปยัง site อื่นๆ 5 site ซึ่งต้องใช้ 1 process สำหรับ 1 รายละเอียด และอาจจะต้องมีการเพิ่ม job process สำหรับล้างพวก transaction Queue ที่ล่าช้า และ user job อื่นๆ สำหรับ snapshot replication แต่ละ snapshot site จะมี 1 รายละเอียดการเชื่อมต่อไปยัง master database และต้องการ job process อย่างน้อยที่สุดเพียง 1 process

นอกจากการกำหนดจำนวนของ job process แล้ว ยังต้องทำการกำหนดค่า job interval ซึ่ง job interval จะเป็นตัวกำหนดความถี่ของ job process ที่จะตื่นขึ้นมา execute operation ที่คอยอยู่ เช่นการ push queue ซึ่งโดยมากกำหนดที่ 60 วินาทีก็เพียงพอสำหรับสิ่งแวดล้อมของการทำ replication แต่ถ้าต้องการให้ประสิทธิภาพสูงขึ้นก็ต้องกำหนดค่าใหม่ได้ขึ้นกับความต้องการเฉพาะงาน เช่น ถ้าต้องการให้มีการกระจายการเปลี่ยนแปลงทุกๆ 20 วินาที ค่า job interval เท่ากับ 60 วินาที คงไม่เพียงพอ

ค่า job process และ job interval ก็กำหนดใน initialization parameter file

job\_queue\_processes = 7

job\_queue\_interval = 60

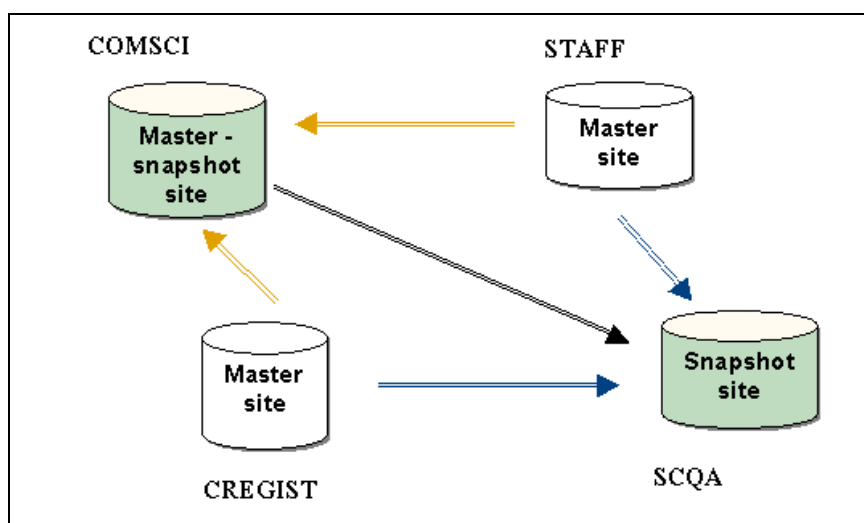
หลังจากที่ทำการเปลี่ยนแปลงค่าแล้วก็ทำการ Restart database อีกครั้ง

### ภาพรวมของการติดตั้ง replication site

หลังจากกำหนดค่า Parameter ที่เหมาะสมแล้ว ขั้นตอนต่อไปคือทำการติดตั้งให้แต่ละ site เหมาะสมกับสิ่งแวดล้อมของการ replication ที่กำหนดไว้

โดยในระบบฐานข้อมูลแบบกระจายของระบบประกันคุณภาพการศึกษาคณะวิทยาศาสตร์ ใช้ฐานข้อมูลทั้งหมด 4 ฐานข้อมูล

- COMSCI
- STAFF
- CREGIST
- SCQA



ภาพประกอบ ก-1 แสดงสิ่งแวดล้อมของการทำสำเนาข้อมูล

จากรูปจะเห็นว่า มี 3 site ที่มีสิ่งแวดล้อมเป็น Master site และ 2 site ที่เป็น snapshot Site

## การติดตั้ง Master Sites

```
/******
```

ขั้นตอนที่ 1 @ STAFF:

```
CONNECT AS SYSTEM AT MASTER SITE
```

```
*****/
```

ทำการเชื่อมต่อกับฐานข้อมูล ติดต่อโดยผู้ใช้งานเป็น system ของ site ที่ต้องการ ติดตั้ง และทำขั้นตอนอื่นๆ กันนี้กับ COMSCI และ CREGIST

```
CONNECT system/manager@STAFF
```

```
/******
```

ขั้นตอนที่ 2 @ STAFF:

```
CREATE REPLICATION ADMINISTRATOR
```

```
*****/
```

สร้างผู้ใช้งาน replication administrator โดยให้สิทธิ์ที่ใช้ในการสร้างและจัดการสิ่งแวดล้อมของการ replication ซึ่งจะต้องสร้าง replication administrator ของแต่ละ database ในสิ่งแวดล้อมของการทำ replication

```
CREATE USER repadmin IDENTIFIED BY repadmin;
```

```
/******
```

ขั้นตอนที่ 3 @ STAFF:

```
GRANT PRIVILEGES TO REPLICATION ADMINISTRATOR
```

```
*****/
```

ทำการ Execute GRANT\_ADMIN\_ANY\_SCHEMA API เพื่อให้สิทธิ์ต่างๆ แก่ replication administrator ในการสร้างและจัดการสิ่งแวดล้อมของการ Replication

```
BEGIN
```

```
DBMS\_REPCAT\_ADMIN.GRANT\_ADMIN\_ANY\_SCHEMA \(  
    USERNAME => 'repadmin');
```

```
END;
```

```
/
```

```
/******
```

ขั้นตอนที่ 4 @ STAFF:

```
REGISTER PROPAGATOR
```

```
*****/
```

ผู้ใช้งาน Propagator มีหน้าที่รับผิดชอบเกี่ยวกับการกระจาย transaction queue ไปยัง master site อื่นๆ

```
BEGIN
```

```
DBMS\_DEFER\_SYS.REGISTER\_PROPAGATOR \(  
    USERNAME => 'repadmin');
```

```
END;
```

```
/
```

```
/******
```

```
ขั้นตอนที่ 5 @ STAFF:
```

```
REGISTER RECEIVER
```

```
*****/
```

ผู้ใช้ Receiver ทำหน้าที่รับ transaction ที่ส่งมาจาก propagator จาก master site อื่น

```
BEGIN
```

```
DBMS_REPCAT_ADMIN.REGISTER_USER_REPGROUP (
```

```
  USERNAME => 'reppadmin',
```

```
  privilege_type => 'receiver',
```

```
  list_of_gnames => NULL);
```

```
END;
```

```
/
```

```
/******
```

```
ขั้นตอนที่ 6 @ STAFF:
```

```
SCHEDULE PURGE AT MASTER SITE
```

```
*****/
```

เพื่อที่จะเก็บขนาดของ Transaction queue ไว้เพื่อการตรวจสอบ เราจึงควรลบ transaction ที่ทำงานเสร็จสมบูรณ์ออกไป โดยใช้ SCHEDULE\_PURGE API ซึ่งจะทำให้การล้าง process เหล่านี้ให้ได้อย่างอัตโนมัติ ดังนั้น replication administrator ต้องทำการ execute procedure นี้

```
CONNECT reppadmin/reppadmin@STAFF
```

```
BEGIN
```

```
DBMS_DEFER_SYS.SCHEDULE_PURGE (
```

```
  next_date => SYSDATE,
```

```
  interval => 'sysdate + 1/24',
```

```
  delay_seconds => 0,
```

```
  rollback_segment => "");
```

```
END;
```

```
/
```

```
/******
```

```
ขั้นตอนที่ 7: CREATE MASTER SITE USERS
```

```
*****/
```

```
--ขั้นตอนที่ 7a: ทำการสร้าง User: PROXY SNAPSHOT ADMINISTRATOR
```

proxy snapshot administrator จะดำเนินการทำงาน ที่ Master site ในนามของ snapshot administrator ของ snapshot site

```
CONNECT system/manager@STAFF
```

```
CREATE USER proxy_snapadmin IDENTIFIED BY proxy_snapadmin;
```

```
BEGIN
```

```
DBMS_REPCAT_ADMIN.REGISTER_USER_REPGROUP (
```

```
  username => 'PROXY_SNAPADMIN',
```

```
  privilege_type => 'PROXY_SNAPADMIN',
```

```
  list_of_gnames => NULL);
```

```
END;
```

```
/
```

--ขั้นตอนที่ 7b: สร้าง PROXY REFRESHER

proxy refresher ทำงานในนามของ refresher ที่ Snapshot site

```
CREATE USER proxy_refresher IDENTIFIED BY proxy_refresher;
```

```
GRANT CREATE SESSION TO proxy_refresher;
GRANT SELECT ANY TABLE TO proxy_refresher;
```

### การติดตั้ง Snapshot Sites

```
/******
```

ขั้นตอนที่ 1:CONNECT AS SYSTEM AT SNAPSHOT SITE

```
*****/
```

ทำการ Connect เป็น system ยัง database ที่ต้องการ set up เป็น snapshot site

```
CONNECT system/manager@scqa
```

```
/******
```

ขั้นตอนที่ 2:CREATE SNAPSHOT SITE USERS

```
*****/
```

มีผู้ใช้ ที่จำเป็นต้องสร้างที่ snapshot site ดังต่อไปนี้

```
-- SNAPSHOT ADMINISTRATOR
```

```
-- PROPAGATOR
```

```
-- REFRESHER
```

--ขั้นตอนที่ 2a: ทำการสร้างผู้ใช้ SNAPSHOT ADMINISTRATOR

snapshot administrator รับผิดชอบเกี่ยวกับการสร้างและจัดการ Snapshot site ซึ่งเราต้องทำการ execute GRANT\_ADMIN\_ANY\_SCHEMA procedure เพื่อทำการให้สิทธิ์ที่เหมาะสมให้ snapshot administrator

```
create user SNAPADMIN identified by SNAPADMIN;
```

```
BEGIN
```

```
DBMS\_REPCAT\_ADMIN.GRANT\_ADMIN\_ANY\_SCHEMA (  
username => 'SNAPADMIN');
```

```
END;
```

```
/
```

--ขั้นตอนที่ 2b: ทำการสร้างผู้ใช้ PROPAGATOR

Propagator ทำหน้าที่เกี่ยวกับการกระจาย transaction queue ไปยัง master site

```
CREATE USER propagator IDENTIFIED BY propagator;
```

```
BEGIN
```

```
DBMS_DEFER_SYS.REGISTER_PROPAGATOR (
```

```
  username => 'propagator');
```

```
END;
```

```
/
```

--ขั้นตอนที่ 2c: สร้างผู้ใช้ REFRESHER

Refresher ทำหน้าที่ดึงข้อมูลที่มีการเปลี่ยนแปลงที่ table ที่ master site มายัง snapshot site

```
create user REFRESHER identified by REFRESHER;
```

```
GRANT CREATE SESSION TO refresher;
```

```
GRANT ALTER ANY SNAPSHOT TO refresher;
```

```
/******
```

```
ขั้นตอนที่ 3:CREATE DATABASE LINKS TO MASTER SITE
```

```
*****/
```

--ขั้นตอนที่ 3A: สร้าง PUBLIC DATABASE LINK ไปยัง Master Site

```
CONNECT system/manager@SCQA
```

```
CREATE PUBLIC DATABASE LINK STAFF.US.ORACLE.COM USING  
'STAFF.US.ORACLE.COM';
```

--ขั้นตอนที่ 3b: CREATE SNAPSHOT ADMINISTRATOR DATABASE LINK

เราจำเป็นต้องสร้าง Database link จาก snapshot administrator ที่ snapshot site ไปยัง proxy snapshot administrator ที่ master site

```
CONNECT snapadmin/snapadmin@scqa;
```

```
CREATE DATABASE LINK STAFF.US.ORACLE.COM  
CONNECT TO proxy_snapadmin IDENTIFIED BY proxy_snapadmin;
```

--ขั้นตอนที่ 3c: สร้าง User: PROPAGATOR/RECEIVER DATABASE LINK

เราจำเป็นต้องสร้าง database link จาก propagator ที่ snapshot site ไปยัง receiver ที่ master site โดย receiver จะถูกกำหนดเมื่อมีการสร้าง master group

```
CONNECT propagator/propagator@scqa
```

```
CREATE DATABASE LINK STAFF.US.ORACLE.COM
CONNECT TO repadmin IDENTIFIED BY repadmin;
```

```
/******
```

```
ขั้นตอนที่ 4: SCHEDULE PURGE AT SNAPSHOT SITE
```

```
*****/
```

เพื่อที่จะเก็บขนาดของ Transaction queue ไว้เพื่อการตรวจสอบ เราจึงควรลบ transaction ที่ทำงานเสร็จสมบูรณ์ออกไป โดยใช้ SCHEDULE\_PURGE API ซึ่งจะทำการล้าง process เหล่านั้นให้ได้อย่างอัตโนมัติ แต่ถ้า snapshot site ทำ snapshot แบบ read-only ก็ไม่ต้องทำการ execute procedure นี้

```
CONNECT snapadmin/snapadmin@scqa
```

```
BEGIN
```

```
DBMS\_DEFER\_SYS.SCHEDULE\_PURGE (
```

```
next_date => SYSDATE,
```

```
interval => 'sysdate + 1/24',
```

```
delay_seconds => 0,
```

```
rollback_segment => "");
```

```
END;
```

```
/
```

```
/******
```

```
ขั้นตอนที่ 5: SCHEDULE PUSH AT SNAPSHOT SITE
```

```
*****/
```

SCHEDULE\_PUSH เป็นตารางรายละเอียดที่ใช้เพื่อจัดการการกระจาย transaction queue ไปยัง Master site ต่างๆ

```
CONNECT snapadmin/snapadmin@scqa
```

```
BEGIN
```

```
DBMS\_DEFER\_SYS.SCHEDULE\_PUSH (
```

```
destination => 'STAFF.US.ORACLE.COM',
```

```
interval => 'sysdate + 1/24',
```

```
next_date => SYSDATE,
```

```
stop_on_error => FALSE,
```

```
delay_seconds => 0,
```

```
parallelism => 0);
```

```
END;
```

```
/
```

## การสร้าง Master group

หลังจากที่ทำการ set up master site (STAFF) เรียบร้อยแล้วก็พร้อมที่จะทำการสร้าง master group ในตัวอย่างนี้จะทำการสร้าง STAFF\_MG ซึ่งเป็น master group ที่ประกอบด้วย object ที่จะถูก replication ไปยัง snapshot site โดยมี schema ADMIN\_AP เป็นเจ้าของ ซึ่งในที่นี่จะใช้ object ดังต่อไปนี้

- STAFF
- STAFF\_EDUCATION
- STAFF\_ONSTUDY
- STAFF\_ACADEMIC

## Create Master Group

```

/*****
ขั้นตอนที่ 1:CREATE MASTER GROUP
*****/

ใช้ CREATE_MASTER_REPGROUP API เพื่อกำหนด master group ใหม่ ซึ่งเราจะทำการตั้ง
ชื่อ master group ที่ขั้นตอนนี้
CONNECT repadmin/repadmin@STAFF

BEGIN
  DBMS_REPCAT.CREATE_MASTER_REPGROUP (
    gname => 'STAFF_MG');
END;
/

/*****
ขั้นตอนที่ 2:ADD OBJECTS TO MASTER GROUP
*****/

ใช้ CREATE_MASTER_REPOBJECT API เพื่อที่เพิ่ม Object เช่น table ไปยัง master group

BEGIN
  DBMS_REPCAT.CREATE_MASTER_REPOBJECT (
    gname => 'STAFF_MG',
    type => 'TABLE',
    oname => 'STAFF',
    sname => 'ADMIN_AP',
    use_existing_object => TRUE,
    copy_rows => TRUE);
END;
/

BEGIN
  DBMS_REPCAT.CREATE_MASTER_REPOBJECT (
    gname => 'STAFF_MG',
    type => 'TABLE',

```



```

        oname => 'STAFF_EDUCATION',
        sname => 'ADMIN_AP',
        use_existing_object => TRUE,
        copy_rows => TRUE);
END;
/

BEGIN
  DBMS\_REPCAT.CREATE\_MASTER\_REPOBJECT (
    gname => 'STAFF_MG',
    type => 'TABLE',
    oname => 'STAFF_ONSTUDY',
    sname => 'ADMIN_AP',
    use_existing_object => TRUE,
    copy_rows => TRUE);
END;
/

BEGIN
  DBMS\_REPCAT.CREATE\_MASTER\_REPOBJECT (
    gname => 'STAFF_MG',
    type => 'TABLE',
    oname => 'STAFF_ACADEMIC',
    sname => 'ADMIN_AP',
    use_existing_object => TRUE,
    copy_rows => TRUE);
END;
/

/*****
๒
ขั้นตอนที่ 3:GENERATE REPLICATION SUPPORT
*****/

BEGIN
  DBMS\_REPCAT.GENERATE\_REPLICATION\_SUPPORT (
    sname => 'ADMIN_AP'
    oname => 'STAFF',
    type => 'TABLE',
    min_communication => TRUE);
END;
/

BEGIN
  DBMS\_REPCAT.GENERATE\_REPLICATION\_SUPPORT (
    sname => 'ADMIN_AP'
    oname => 'STAFF_EDUCATION',
    type => 'TABLE',
    min_communication => TRUE);
END;
/

BEGIN
  DBMS\_REPCAT.GENERATE\_REPLICATION\_SUPPORT (
    sname => 'ADMIN_AP'
    oname => 'STAFF_ONSTUDY',

```

```

    type => 'TABLE',
    min_communication => TRUE);
END;
/

```

```

BEGIN
  DBMS\_REPCAT.GENERATE\_REPLICATION\_SUPPORT (
    sname => 'ADMIN_AP'
    oname => 'STAFF_ACDEMIC',
    type => 'TABLE',
    min_communication => TRUE);
END;
/

```

```

/*****
ขั้นตอนที่ 4:RESUME REPLICATION
*****/

```

ทำการ Resume replication activity โดยใช้ [RESUME MASTER ACTIVITY procedure](#) API

```

BEGIN
  DBMS\_REPCAT.RESUME\_MASTER\_ACTIVITY (
    gname => 'STAFF_MG');
END;
/

```

### การสร้าง Snapshot group

หลังจากทำการ Set up snapshot site และทำการสร้าง master group เรียบร้อยแล้ว

ขั้นต่อไปก็ทำการสร้าง snapshot group ที่ snapshot site

```

/*****
ขั้นตอนที่ 1:CREATE SNAPSHOT LOGS AT MASTER SITE
*****/

```

ทำการสร้าง snapshot log ที่ master site

ต้องทำการสร้าง snapshot log สำหรับ master table ที่จะ replication มายัง snapshot site

```
CONNECT admin_ap@staff
```

```

CREATE SNAPSHOT LOG ON admin_ap.staff;
CREATE SNAPSHOT LOG ON admin_ap.staff_education;
CREATE SNAPSHOT LOG ON admin_ap.staff_onstudy;
CREATE SNAPSHOT LOG ON admin_ap.staff_academic;

```

```

/*****
ขั้นตอนที่ 2:CREATE REPLICATED SCHEMA AND LINKS
*****/

```

ผู้ใช้ admin\_ap บนฐานข้อมูล SCQA เป็นเจ้าของ snapshot ที่ replicate มาจาก master site

```
CONNECT admin\_ap@SCQA
```

จากนั้นสร้าง Database link ไปยัง proxy\_refresher

```
CREATE DATABASE LINK STAFF.US.ORACLE.COM
CONNECT TO proxy_refresher IDENTIFIED BY proxy_refresher;
```

```
/******
```

```
ขั้นตอนที่ 3:CREATE SNAPSHOT GROUP
```

```
*****/
```

procedure ต่อไปนี้ต้อง execute โดย snapshot administrator ที่ snapshot site และ Master group ที่ระบุใน GNAME ต้องตรงกับที่เราชื่อของ master group ไว้ใน master site

```
CONNECT snapadmin/snapadmin@scqa
```

```
BEGIN
```

```
DBMS\_REPCAT.CREATE\_SNAPSHOT\_REPGROUP (
```

```
  gname => 'STAFF_MG',
  master => 'STAFF.US.ORACLE.COM',
  propagation_mode => 'ASYNCHRONOUS');
```

```
END;
```

```
/
```

```
/******
```

```
ขั้นตอนที่ 4:CREATE REFRESH GROUP
```

```
*****/
```

ทำการสร้าง refresh group เพื่อให้จะให้ snapshot ที่สร้างขึ้น refresh ในเวลาเดียวกันตามกลุ่มที่เหมาะสม และเพื่อให้แน่ใจว่า transaction มีความถูกต้องและตรวจสอบการ refresh ของ snapshot ได้ง่าย จากตัวอย่างนี้ interval => 'sysdate + 1/24' จะทำการ refresh ทุก 1 ชั่วโมง (ถ้าบวก 1 เท่ากับ 1 วัน ถ้า 1/1440 = 1 นาที)

```
BEGIN
```

```
DBMS\_REFRESH.MAKE (
```

```
  name => 'SNAPADMIN.STAFF_RG',
  list => "",
  next_date => SYSDATE,
  interval => 'sysdate + 1/24',
  implicit_destroy => FALSE,
  rollback_seg => "",
  push_deferred_rpc => TRUE,
  refresh_after_errors => FALSE);
```

```
END;
```

```
/
```

```
/******
```

```
ขั้นตอนที่ 5:ADD OBJECTS TO SNAPSHOT GROUP
```

\*\*\*\*\*/

ทำการเพิ่ม object ไปใน snapshot group

BEGIN

DBMS\_REPCAT.CREATE\_SNAPSHOT\_REPOBJECT (

gname => 'STAFF\_MG',

sname => 'ADMIN\_AP',

oname => 'STAFF',

type => 'SNAPSHOT',

ddl\_text => 'create snapshot ADMIN\_AP.STAFF refresh fast with  
primary key as select \* from

ADMIN\_AP.STAFF@STAFF.US.ORACLE.COM',

min\_communication => TRUE);

END;

/

BEGIN

DBMS\_REPCAT.CREATE\_SNAPSHOT\_REPOBJECT (

gname => 'STAFF\_MG',

sname => 'ADMIN\_AP',

oname => 'STAFF\_EDUCATION',

type => 'SNAPSHOT',

ddl\_text => 'create snapshot ADMIN\_AP.STAFF\_EDUCATION refresh fast with  
primary key as select \* from

ADMIN\_AP.STAFF@STAFF\_EDUCATION.US.ORACLE.COM',

min\_communication => TRUE);

END;

/

BEGIN

DBMS\_REPCAT.CREATE\_SNAPSHOT\_REPOBJECT (

gname => 'STAFF\_MG',

sname => 'ADMIN\_AP',

oname => 'STAFF\_ONSTUDY',

type => 'SNAPSHOT',

ddl\_text => 'create snapshot ADMIN\_AP.STAFF\_ONSTUDY refresh fast with  
primary key as select \* from

ADMIN\_AP.STAFF@STAFF\_ONSTUDY.US.ORACLE.COM',

min\_communication => TRUE);

END;

/

BEGIN

DBMS\_REPCAT.CREATE\_SNAPSHOT\_REPOBJECT (

gname => 'STAFF\_MG',

sname => 'ADMIN\_AP',

oname => 'STAFF\_ACADEMIC',

type => 'SNAPSHOT',

ddl\_text => 'create snapshot ADMIN\_AP.STAFF\_ACADEMIC refresh fast with  
primary key as select \* from

ADMIN\_AP.STAFF@STAFF\_ACADEMIC.US.ORACLE.COM',

min\_communication => TRUE);

END;

/

```
/******
```

```
ขั้นตอนที่ 6:ADD OBJECTS TO REFRESH GROUP
```

```
*****/
```

ทำการใส่ object snapshot ที่สร้างขึ้นไว้ใน refresh group ซึ่ง snapshot เหล่านี้จะถูก refresh ตามเวลาที่ระบุในเวลาเดียวกัน

```
BEGIN
```

```
  DBMS\_REFRESH.ADD \(
```

```
    name => 'SNAPADMIN.STAFF_RG',
```

```
    list => 'ADMIN_AP.STAFF',
```

```
    lax => TRUE);
```

```
END;
```

```
/
```

```
BEGIN
```

```
  DBMS\_REFRESH.ADD \(
```

```
    name => 'SNAPADMIN.STAFF_RG',
```

```
    list => 'ADMIN_AP.STAFF_EDUCATION',
```

```
    lax => TRUE);
```

```
END;
```

```
/
```

```
BEGIN
```

```
  DBMS\_REFRESH.ADD \(
```

```
    name => 'SNAPADMIN.STAFF_RG',
```

```
    list => 'ADMIN_AP.STAFF_ONSTUDY',
```

```
    lax => TRUE);
```

```
END;
```

```
/
```

```
BEGIN
```

```
  DBMS\_REFRESH.ADD \(
```

```
    name => 'SNAPADMIN.STAFF_RG',
```

```
    list => 'ADMIN_AP.STAFF_ACADEMIC',
```

```
    lax => TRUE);
```

```
END;
```

```
/
```