**System**

SQL\*Plus: Release 10.2.0.1.0 - Production on Sun Apr 19 18:02:27 2015

Copyright (c) 1982, 2005, Oracle. All rights reserved.

SQL> connect system

Enter password:

Connected.

SQL> create user payroll identified by payroll;

User created.

SQL> create user satyen identified by satyen;

User created.

SQL> create user akash identified by akash;

User created.

SQL> grant all privileges to payroll;

Grant succeeded.

SQL> grant all privileges to satyen;

Grant succeeded.

SQL> grant all privileges to akash;

Grant succeeded.

**Payroll**

SQL> connect payroll

Enter password:

Connected.

SQL> create public database link link\_s connect to satyen identified by satyen using 'XE';

Database link created.

SQL> create public database link link\_a connect to akash identified by akash using 'XE';

Database link created.

SQL> create table salary(basic int not null, IT int, insurance int, bonus int, CONSTRAINT salary\_pk PRIMARY KEY(basic));

Table created.

SQL> create table designation(dname varchar(15) not null, basic int not null, CONSTRAINT designation\_pk PRIMARY KEY(dname), CONSTRAINT fk\_salary FOREIGN KEY(basic) REFERENCES salary(basic));

Table created.

SQL> create table employee(empid int not null, name varchar(20) not null, pno varchar(10) not null, dname varchar(15) not null, CONSTRAINT employee\_pk PRIMARY KEY(empid), CONSTRAINT fk\_designation FOREIGN KEY(dname) REFERENCES designation(dname));

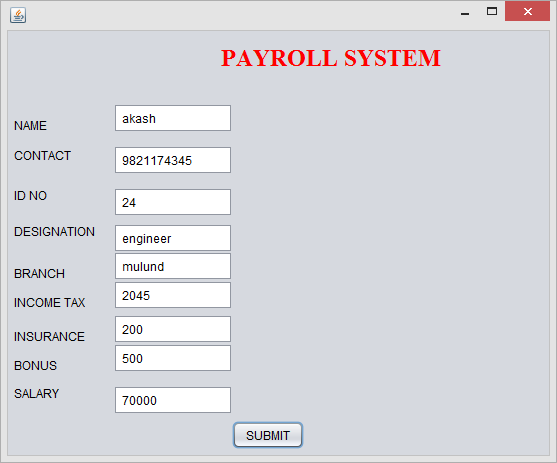
Table created.

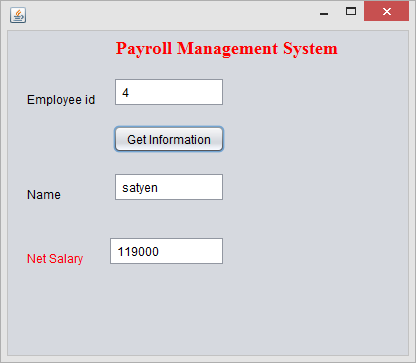
SQL> create table branch(bname varchar(10) not null, empid int not null, CONSTRAINT branch\_pk PRIMARY KEY(bname), CONSTRAINT fk\_employee FOREIGN KEY(empid) REFERENCES employee(empid));

Table created.

SQL> create table store(netsal int, empid int not null, CONSTRAINT store\_pk PRIMARY KEY(netsal), CONSTRAINT fk\_employee1 FOREIGN KEY(empid) REFERENCES employee(empid));

Table created.





**Fragmentation:**

SQL> connect satyen

Enter password:

Connected.

SQL> create table sal1(basic int not null, IT int, insurance int, bonus int, CON

STRAINT salary\_pk PRIMARY KEY(basic));

Table created.

SQL> connect akash

Enter password:

Connected.

SQL> create table sal2(basic int not null, IT int, insurance int, bonus int, CON

STRAINT salary\_pk PRIMARY KEY(basic));

Table created.

SQL> connect payroll

Enter password:

Connected.

SQL> create or replace trigger tsal

2 after insert on salary

3 for each row when(new.basic is NOT NULL)

4 begin if:new.basic>100000 then insert into sal1@link\_s values(:new.basic,:n

ew.IT,:new.insurance,:new.bonus);

5 else insert into sal2@link\_a values(:new.basic,:new.IT,:new.insurance,:new.

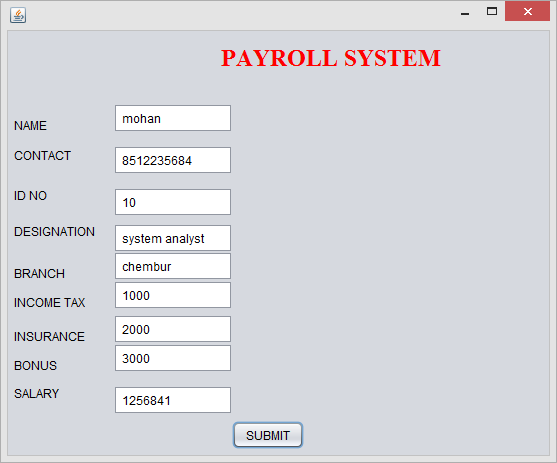
bonus);

6 end if;

7 end;

8 /

Trigger created.



SQL> select \* from employee;

EMPID NAME PNO DNAME

---------- -------------------- ---------- ---------------

4 satyen 9821174634 hr

24 akash 9821174345 engineer

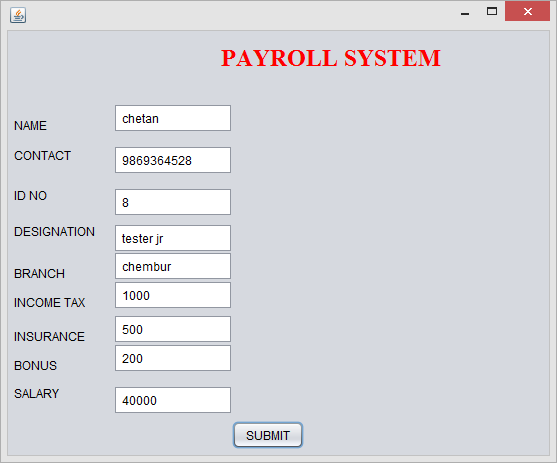
10 mohan 8512235684 system analyst

SQL> select \* from sal1@link\_s;

BASIC IT INSURANCE BONUS

---------- ---------- ---------- ----------

1256841 1000 2000 3000



SQL> select \* from sal2@link\_a;

BASIC IT INSURANCE BONUS

---------- ---------- ---------- ----------

40000 1000 2000 3000

**Synonym:**

SQL> connect satyen

Enter password:

Connected.

SQL> create database link linko connect to payroll identified by payroll using '

XE';

Database link created.

SQL> select \* from employee@linko;

EMPID NAME PNO DNAME

---------- -------------------- ---------- ---------------

4 satyen 9821174634 hr

24 akash 9821174345 engineer

10 mohan 8512235684 system analyst

SQL> create synonym syno for employee@linko;

Synonym created.

SQL> select \* from syno;

EMPID NAME PNO DNAME

---------- -------------------- ---------- ---------------

4 satyen 9821174634 hr

24 akash 9821174345 engineer

10 mohan 8512235684 system analyst

**Query Optimization:**

SQL> select \* from sal1 @link\_s union select \* from sal2 @link\_a;

BASIC IT INSURANCE BONUS

---------- ---------- ---------- ----------

40000 1000 2000 3000

1256841 1000 2000 3000

Execution Plan

----------------------------------------------------------

Plan hash value: 3174456656

--------------------------------------------------------------------------------

---------

| Id | Operation | Name | Rows | Bytes | Cost (%CPU)| Time | Inst

|IN-OUT|

--------------------------------------------------------------------------------

---------

| 0 | SELECT STATEMENT | | 164 | 8528 | 6 (67)| 00:00:01 |

| |

| 1 | SORT UNIQUE | | 164 | 8528 | 6 (67)| 00:00:01 |

| |

| 2 | UNION-ALL | | | | | |

| |

| 3 | REMOTE | SAL1 | 82 | 4264 | 2 (0)| 00:00:01 | LINK\_S

| R->S |

| 4 | REMOTE | SAL2 | 82 | 4264 | 2 (0)| 00:00:01 | LINK\_A

| R->S |

--------------------------------------------------------------------------------

---------

Remote SQL Information (identified by operation id):

----------------------------------------------------

3 - SELECT "BASIC","IT","INSURANCE","BONUS" FROM "SAL1" "SAL1" (accessing

'LINK\_S' )

4 - SELECT "BASIC","IT","INSURANCE","BONUS" FROM "SAL2" "SAL2" (accessing

'LINK\_A' )

Statistics

----------------------------------------------------------

1 recursive calls

0 db block gets

0 consistent gets

0 physical reads

0 redo size

625 bytes sent via SQL\*Net to client

380 bytes received via SQL\*Net from client

2 SQL\*Net roundtrips to/from client

1 sorts (memory)

0 sorts (disk)

2 rows processed

SQL>

SQL> select \* from sal1@link\_s, sal2@link\_a;

BASIC IT INSURANCE BONUS BASIC IT INSURANCE

---------- ---------- ---------- ---------- ---------- ---------- ----------

BONUS

----------

1256841 1000 2000 3000 40000 1000 2000

3000

Execution Plan

----------------------------------------------------------

Plan hash value: 3218680192

--------------------------------------------------------------------------------

-------------

| Id | Operation | Name | Rows | Bytes | Cost (%CPU)| Time | In

st |IN-OUT|

--------------------------------------------------------------------------------

-------------

| 0 | SELECT STATEMENT | | 6724 | 682K| 188 (12)| 00:00:03 |

| |

| 1 | MERGE JOIN CARTESIAN| | 6724 | 682K| 188 (12)| 00:00:03 |

| |

| 2 | REMOTE | SAL1 | 82 | 4264 | 2 (0)| 00:00:01 | LI

NK\_S | R->S |

| 3 | BUFFER SORT | | 82 | 4264 | 186 (12)| 00:00:03 |

| |

| 4 | REMOTE | SAL2 | 82 | 4264 | 2 (0)| 00:00:01 | LI

NK\_A | R->S |

--------------------------------------------------------------------------------

-------------

Remote SQL Information (identified by operation id):

----------------------------------------------------

2 - SELECT "BASIC","IT","INSURANCE","BONUS" FROM "SAL1" "SAL1" (accessing

'LINK\_S' )

4 - SELECT "BASIC","IT","INSURANCE","BONUS" FROM "SAL2" "SAL2" (accessing

'LINK\_A' )

Statistics

----------------------------------------------------------

1 recursive calls

0 db block gets

0 consistent gets

0 physical reads

0 redo size

819 bytes sent via SQL\*Net to client

380 bytes received via SQL\*Net from client

2 SQL\*Net roundtrips to/from client

1 sorts (memory)

0 sorts (disk)

1 rows processed

SQL>

SQL> select \* from salary;

BASIC IT INSURANCE BONUS

---------- ---------- ---------- ----------

120000 2000 1000 2000

70000 2045 200 500

1256841 1000 2000 3000

40000 1000 2000 3000

**DEADLOCK:**

SQL> connect satyen

Enter password:

Connected.

SQL> select \* from sal1;

BASIC IT INSURANCE BONUS

---------- ---------- ---------- ----------

1256841 1000 2000 3000

150000 100 200 300

SQL> update sal1 set IT='200' where insurance='200';

1 row updated.

SQL> update sal1 set IT='200' where insurance='2000';

SQL> connect payroll

Enter password:

Connected.

SQL> select \* from satyen.sal1;

BASIC IT INSURANCE BONUS

---------- ---------- ---------- ----------

1256841 1000 2000 3000

150000 100 200 300

SQL> update satyen.sal1 set IT='100' where insurance='2000';

1 row updated.

SQL> update satyen.sal1 set IT='100' where insurance='200';

update satyen.sal1 set IT='100' where insurance='200'

\*

ERROR at line 1:

ORA-00060: deadlock detected while waiting for resource