EX.NO.: 11		PL SQL PROGRAMS
DATE :	24/09/2024	

PROGRAM 1
WRITE A PL/SQL BLOCK TO CALCULATE THE INCENTIVE OF AN EMPLOYEE WHOSE ID IS 110.

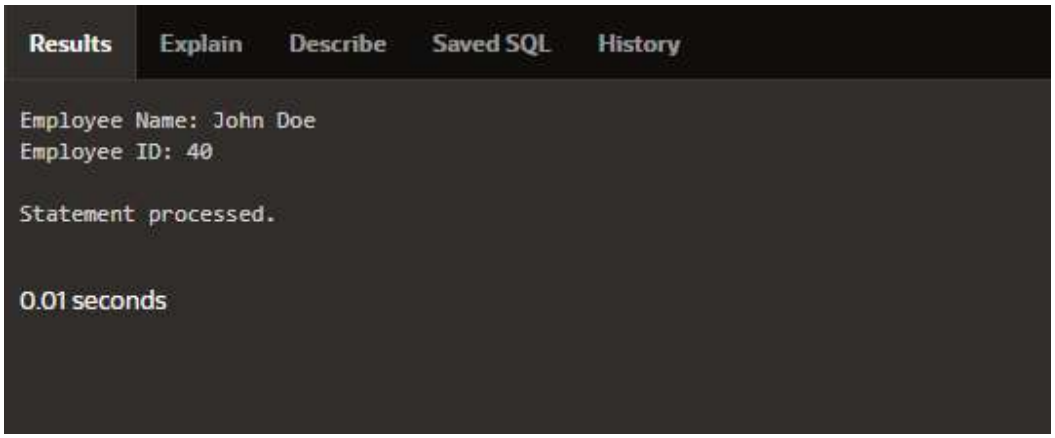
```
DECLARE
  PL_EMP_ID EMPLOYEES.EMPLOYEE_ID%TYPE
  := 110;PL_SALARY EMPLOYEES.SALARY%TYPE;
  PL_INCENTIVE
NUMBER;BEGIN
  SELECT SALARY INTO
  PL_SALARYFROM
  EMPLOYEES
  WHERE EMPLOYEE_ID = PL_EMP_ID;

  PL_INCENTIVE := PL_SALARY

  * 0.10;UPDATE EMPLOYEES
  SET INCENTIVE = PL_INCENTIVE
  WHERE EMPLOYEE_ID =
  PL_EMP_ID;

  DBMS_OUTPUT.PUT_LINE('INCENTIVE FOR EMPLOYEE ID ' || PL_EMP_ID ||
' IS ' ||PL_INCENTIVE);

  COMMIT;
END;
```

Results | Explain | Describe | Saved SQL | History

```
Incentive for employee ID 110 is 820

1 row(s) updated.

0.00 seconds
```

PROGRAM 2
WRITE A PL/SQL BLOCK TO SHOW AN INVALID CASE-INSENSITIVE REFERENCE TO A QUOTEDAND WITHOUT QUOTED USER-DEFINED IDENTIFIER.

```
DECLARE
  EMPLOYEENAME VARCHAR2(100);
  "EMPLOYEEID"  NUMBER;
BEGIN
  EMPLOYEENAME := 'JOHN
  DOE';"EMPLOYEEID" := 40;

  DBMS_OUTPUT.PUT_LINE('EMPLOYEE NAME: ' ||
  EMPLOYEENAME);DBMS_OUTPUT.PUT_LINE('EMPLOYEE ID: ' ||
  "EMPLOYEEID");
END;
```



PROGRAM 3

WRITE A PL/SQL BLOCK TO ADJUST THE SALARY OF THE EMPLOYEE WHOSE ID 122.SAMPLE TABLE: EMPLOYEES

```
DECLARE
   V_EMPLOYEE_ID NUMBER :=
   122;V_SALARY   NUMBER;
   V_NEW_SALARY  NUMBER;
   V_INCREASE_PERCENTAGE NUMBER
:= 0.40;BEGIN
   SELECT SALARY INTO
   V_SALARYFROM
   EMPLOYEES
   WHERE EMPLOYEE_ID = V_EMPLOYEE_ID;

   V_NEW_SALARY := V_SALARY + (V_SALARY *

   V_INCREASE_PERCENTAGE / 100);UPDATE EMPLOYEES
   SET SALARY = V_NEW_SALARY
   WHERE EMPLOYEE_ID = V_EMPLOYEE_ID;

   DBMS_OUTPUT.PUT_LINE('EMPLOYEE ID ' || V_EMPLOYEE_ID || ' NEW
SALARY: ' ||V_NEW_SALARY);
END;
```

Results | Explain | Describe | Saved SQL | History

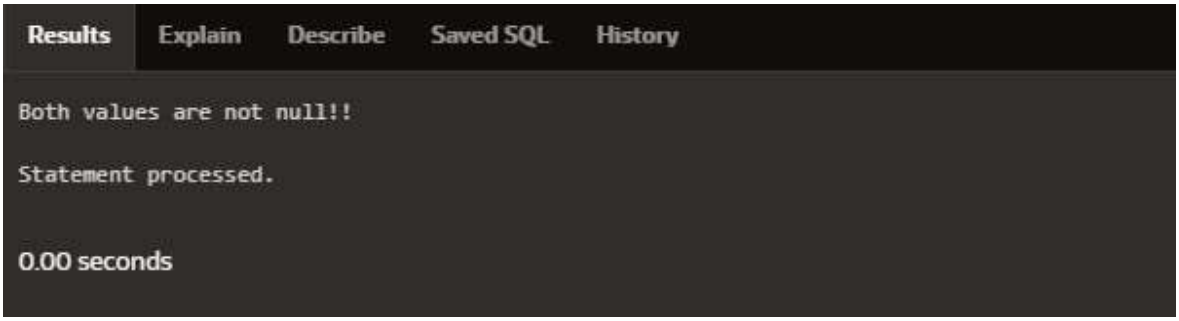
```
Employee ID 122 new salary: 9036.036

1 row(s) updated.
```

0.01 seconds

PROGRAM 4

WRITE A PL/SQL BLOCK TO CREATE A PROCEDURE USING THE "IS [NOT] NULL OPERATOR" ANDSHOW AND OPERATOR RETURNS TRUE IF AND ONLY IF BOTH OPERANDS ARE TRUE.

```
CREATE OR REPLACE PROCEDURE
CHECK_NULLIS
   VALUE1 NUMBER :=
   10; VALUE2 NUMBER
   := NULL;
BEGIN
   IF VALUE1 IS NOT NULL AND VALUE2 IS NULL THEN
      DBMS_OUTPUT.PUT_LINE('BOTH VALUES ARE NOT
      NULL!!');
   ELSE
      DBMS_OUTPUT.PUT_LINE('NULL VALUE
   FOUND');END IF;
END;


BEGIN
   CHECK_NU
LL;END;
```



PROGRAM 5

WRITE A PL/SQL BLOCK TO DESCRIBE THE USAGE OF LIKE OPERATOR INCLUDING WILDCARDCHARACTERS AND ESCAPE CHARACTER.

```
DECLARE

V_EMPLOYEENAME
EMPLOYEES.FIRST_NAME%TYPE;
V_EMPLOYEEID NUMBER := 122;

BEGIN

SELECT FIRST_NAME INTO
V_EMPLOYEENAMEFROM
EMPLOYEES
WHERE FIRST_NAME LIKE '%E%' AND EMPLOYEE_ID =

V_EMPLOYEEID;

DBMS_OUTPUT.PUT_LINE(V_EMPLOYEENAME);

END;
```

PROGRAM 6

WRITE A PL/SQL PROGRAM TO ARRANGE THE NUMBER OF TWO VARIABLE IN SUCH A WAY THAT THE SMALL NUMBER WILL STORE IN NUM_SMALL VARIABLE AND LARGE NUMBER WILL STORE IN NUM_LARGEVARIABLE.

```
DECLARE
AB NUMBER :=10;
CD NUMBER :=20;
NUM_SMALL
NUMBER;
NUM_LARGE
NUMBER;BEGIN
IF AB>CD THEN
NUM_SMALL
:=CD;
NUM_LARGE
:=AB;ELSE
NUM_SMALL
:=AB;
NUM_LARGE
:=CD;END IF;
DBMS_OUTPUT.PUT_LINE('SMALL NUMBER =
'||NUM_SMALL);DBMS_OUTPUT.PUT_LINE('LARGE
NUMBER = '||NUM_LARGE); END;
```

```
small number = 10
large number = 20

Statement processed.

0.01 seconds
```

PROGRAM 7

WRITE A PL/SQL PROCEDURE TO CALCULATE THE INCENTIVE ON A TARGET ACHIEVED AND DISPLAY THEMESSAGE EITHER THE RECORD UPDATED OR NOT.

```
CREATE OR REPLACE PROCEDURE
CALCULATE_INCENTIVE(P_EMP_ID
EMPLOYEES.EMPLOYEE_ID%TYPE, P_TARGET
NUMBER)
IS
   V_INCENTIVE  NUMBER(7,2);
   V_SALARY
   EMPLOYEES.SALARY%TYPE;
BEGIN
   SELECT SALARY INTO
   V_SALARYFROM
   EMPLOYEES
   WHERE EMPLOYEE_ID = P_EMP_ID;

   IF P_TARGET >= 100000 THEN
      V_INCENTIVE := V_SALARY
      * 0.1;
      DBMS_OUTPUT.PUT_LINE('INCENTIVE OF ' || V_INCENTIVE || ' CALCULATED FOR
EMPLOYEE ID ' ||P_EMP_ID);
   ELSE
      DBMS_OUTPUT.PUT_LINE('NO INCENTIVE FOR EMPLOYEE ID ' ||
   P_EMP_ID);END IF;
END;
```

```
Incentive of 750 calculated for employee ID 176

Statement processed.


0.02 seconds
```

PROGRAM 8

WRITE A PL/SQL PROCEDURE TO CALCULATE INCENTIVE ACHIEVED ACCORDING TO THE SPECIFIC SALELIMIT.

```
CREATE OR REPLACE PROCEDURE INCENTIVE_SALE(P_EMP_ID
EMPLOYEES.EMPLOYEE_ID%TYPE,P_SALES NUMBER)
IS
   V_INCENTIVE
NUMBER(7,2);BEGIN
   IF P_SALES > 100000 THEN
      V_INCENTIVE := P_SALES *
      0.1;
   ELSIF P_SALES BETWEEN 50000 AND
      100000 THENV_INCENTIVE := P_SALES *
      0.05;
   ELSE
      V_INCENTIVE :=
   0;END IF;

   DBMS_OUTPUT.PUT_LINE('INCENTIVE FOR EMPLOYEE ID ' || P_EMP_ID || ' IS: ' ||
V_INCENTIVE);END;



BEGIN
   INCENTIVE_SALE(122,5000
00);END;
```

```
Incentive for employee ID 122 is: 50000

Statement processed.

0.01 seconds
```

PROGRAM 9

WRITE A PL/SQL PROGRAM TO COUNT NUMBER OF EMPLOYEES IN DEPARTMENT 50 AND CHECKWHETHER THIS DEPARTMENT HAVE ANY VACANCIES OR NOT. THERE ARE 45 VACANCIES IN THIS DEPARTMENT.

```
DECLARE
NO_OF_EMP NUMBER;
VACANCIES
NUMBER:=45;BEGIN
SELECT COUNT(*) INTO NO_OF_EMP FROM EMPLOYEES WHERE
DEPARTMENT_ID=50;IF NO_OF_EMP<VACANCIES THEN
DBMS_OUTPUT.PUT_LINE('VACANCIES ARE
AVAILABLE');ELSE
DBMS_OUTPUT.PUT_LINE('VACANCIES ARE NOT
AVAILABLE');END IF;
END;
```

```
vacancies are available

Statement processed.

0.01 seconds
```

PROGRAM 10

WRITE A PL/SQL PROGRAM TO COUNT NUMBER OF EMPLOYEES IN A SPECIFIC DEPARTMENT AND CHECK WHETHER THIS DEPARTMENT HAVE ANY VACANCIES OR NOT. IF ANY VACANCIES, HOW MANYVACANCIES ARE IN THAT DEPARTMENT.

```
DECLARE
   V_DEPARTMENT_ID NUMBER
   := 55;V_EMP_COUNT
   NUMBER; V_VACANCIES
   NUMBER := 50;
BEGIN
   SELECT COUNT(*) INTO
   V_EMP_COUNTFROM
   EMPLOYEES
   WHERE DEPARTMENT_ID = V_DEPARTMENT_ID;

   IF V_EMP_COUNT < V_VACANCIES THEN
      DBMS_OUTPUT.PUT_LINE('VACANCIES AVAILABLE: ' || (V_VACANCIES -
   V_EMP_COUNT));ELSE
      DBMS_OUTPUT.PUT_LINE('NO VACANCIES
   AVAILABLE.');END IF;
END;
```

```
Vacancies available: 47

Statement processed.

0.01 seconds
```

PROGRAM 11

WRITE A PL/SQL PROGRAM TO DISPLAY THE EMPLOYEE IDS, NAMES, JOB TITLES, HIRE DATES, ANDSALARIES OF ALL EMPLOYEES.

```
BEGIN
   FOR I IN (SELECT EMPLOYEE_ID, FIRST_NAME || ' ' || LAST_NAME AS NAME,
JOB_ID, HIRE_DATE,SALARY FROM EMPLOYEES)
   LOOP
      DBMS_OUTPUT.PUT_LINE('ID: ' || I.EMPLOYEE_ID || ', NAME: ' || I.NAME || ', JOB: ' ||
      I.JOB_ID
|| ', HIRE DATE: ' || I.HIRE_DATE || ', SALARY: ' ||
   I.SALARY);END LOOP;
END;
```

```
ID: 2, Name: Emma Austen, Job: ST_CLERK, Hire Date: 11/06/1990, Salary: 5500
ID: 10, Name: Paul Rudd, Job: #pr010, Hire Date: 04/06/1969, Salary: 2500
ID: 11, Name: Brie Zlotkey, Job: #bl011, Hire Date: 10/01/1989, Salary: 7200
ID: 20, Name: Elizabeth Olsen, Job: #eo020, Hire Date: 02/16/1989, Salary: 7300
ID: 25, Name: Cate Abu, Job: #cb025, Hire Date: 05/14/1969, Salary: 13500
ID: 27, Name: Jeff Goldblum, Job: ST_CLERK, Hire Date: 10/22/1952, Salary: 3500
ID: 122, Name: Robert Downey, Job: #rd003, Hire Date: 04/04/1965, Salary: 9036.04
ID: 18, Name: Karen Gillan, Job: #kg018, Hire Date: 11/28/1987, Salary: 6900
ID: 21, Name: Anthony Mackie, Job: ST_CLERK, Hire Date: 09/23/1978, Salary: 4000
ID: 22, Name: Sebastian Stan, Job: #ss022, Hire Date: 08/13/1982, Salary: 9000
ID: 28, Name: Karl Austin, Job: #ka028, Hire Date: 06/07/1972, Salary: 13500
ID: 176, Name: Chris Morris, Job: #ce005, Hire Date: 05/07/1994, Salary: 7500
ID: 6, Name: Mark Ruffalo, Job: #mr006, Hire Date: 11/22/1967, Salary: 7200
ID: 12, Name: Chadwick Boseman, Job: #cb012, Hire Date: 11/29/1976, Salary: 8000
ID: 24, Name: Tom Hiddleston, Job: #th024, Hire Date: 02/09/1981, Salary: 6500
ID: 1, Name: Justin Beiber, Job: ST_CLERK, Hire Date: 09/21/1996, Salary: 4900
ID: 8, Name: Jeremy Wilson, Job: #ja008, Hire Date: 01/07/1971, Salary: 13500
ID: 7, Name: Chris Hemsworth, Job: #ch007, Hire Date: 08/11/1983, Salary: 7800
ID: 9, Name: Tom Holland, Job: ST_CLERK, Hire Date: 06/01/1996, Salary: 6000
ID: 13, Name: Chris Austin, Job: #ca013, Hire Date: 06/21/1979, Salary: 13500
ID: 17, Name: Dave Bautista, Job: #db017, Hire Date: 01/18/1969, Salary: 6500
ID: 26, Name: Tessa Thompson, Job: ST_CLERK, Hire Date: 10/03/1983, Salary: 5200
ID: 14, Name: Zoe Austin, Job: #za014, Hire Date: 06/19/1978, Salary: 13500
ID: 19, Name: Pom Davies, Job: #pk019, Hire Date: 05/03/1986, Salary: 1100
ID: 42, Name: Matos roy, Job: #mr042, Hire Date: 02/23/1991, Salary: 7000
ID: 4, Name: Scarlett Austin, Job: #sa004, Hire Date: 11/22/1984, Salary: 13500
ID: 15, Name: Bradley Hook, Job: ST_CLERK, Hire Date: 01/05/1975, Salary: 4500
ID: 16, Name: Vin Diesel, Job: #vd016, Hire Date: 07/18/1967, Salary: 8000
ID: 110, Name: Benedict andru, Job: #bc023, Hire Date: 07/19/1976, Salary: 8200
ID: 30, Name: Taika Waititi, Job: #tw030, Hire Date: 08/16/1975, Salary: 7700
ID: 40, Name: John Doe , Job:   #jd040 , Hire Date: 08/10/1995, Salary: 6000
ID: 29, Name: Idris Elba, Job: #ie029, Hire Date: 09/06/1972, Salary: 7400
ID: 41, Name: Matos charles, Job: #mc041, Hire Date: 09/18/1993, Salary: 8900

Statement processed.
```

PROGRAM 12

WRITE A PL/SQL PROGRAM TO DISPLAY THE EMPLOYEE IDS, NAMES, AND DEPARTMENT NAMES OFALL EMPLOYEES.

```
BEGIN
   FOR I IN (SELECT E.EMPLOYEE_ID, E.FIRST_NAME || ' ' || E.LAST_NAME AS NAME,
            D.DEPT_NAMEFROM EMPLOYEES E
            JOIN DEPARTMENT D ON E.EMPLOYEE_ID = D.DEPT_ID) LOOP
      DBMS_OUTPUT.PUT_LINE('ID: ' || I.EMPLOYEE_ID || ', NAME: ' || I.NAME || ',
      DEPARTMENT: ' ||
I.DEPT_NAM
   E);END
   LOOP;
END;
```

```
ID: 25, Name: Cate Abu, Department: executive
ID: 15, Name: Bradley Hook, Department: sales manager
ID: 30, Name: Taika Waititi, Department: accounts manager

Statement processed.

0.03 seconds
```

PROGRAM 13

WRITE A PL/SQL PROGRAM TO DISPLAY THE JOB IDS, TITLES, AND MINIMUM SALARIES OF ALL JOBS.

```
BEGIN
   FOR REC IN (SELECT E.EMPLOYEE_ID, D.DEPT_NAME, MIN(SALARY) AS
MIN_SALARY FROMEMPLOYEES
   E JOIN DEPARTMENT D
   ON E.EMPLOYEE_ID = D.DEPT_ID
   GROUP BY E.EMPLOYEE_ID ,
   D.DEPT_NAME)LOOP
      DBMS_OUTPUT.PUT_LINE('JOB ID: ' || REC.EMPLOYEE_ID || ', TITLE: ' ||
REC.DEPT_NAME || ',MIN SALARY: ' || REC.MIN_SALARY);
   END
LOOP;END;
```

```
Job ID: 30, Title: accounts manager, Min Salary: 7700
Job ID: 25, Title: executive, Min Salary: 13500
Job ID: 15, Title: sales manager, Min Salary: 4500

Statement processed.

0.05 seconds
```

WRITE A PL/SQL PROGRAM TO DISPLAY THE JOB IDS, TITLES, AND MINIMUM SALARIES OF ALL JOBS.

```
BEGIN
   FOR REC IN (SELECT E.EMPLOYEE_ID, D.DEPT_NAME, MIN(SALARY) AS
MIN_SALARY FROMEMPLOYEES
   E JOIN DEPARTMENT D
   ON E.EMPLOYEE_ID = D.DEPT_ID
   GROUP BY E.EMPLOYEE_ID ,
   D.DEPT_NAME)LOOP
      DBMS_OUTPUT.PUT_LINE('JOB ID: ' || REC.EMPLOYEE_ID || ', TITLE: ' ||
REC.DEPT_NAME || ',MIN SALARY: ' || REC.MIN_SALARY);
   END
LOOP;END;
```

```
Job ID: 30, Title: accounts manager, Min Salary: 7700
Job ID: 25, Title: executive, Min Salary: 13500
Job ID: 15, Title: sales manager, Min Salary: 4500

Statement processed.

0.05 seconds
```

PROGRAM 14
WRITE A PL/SQL PROGRAM TO DISPLAY THE EMPLOYEE IDS, NAMES, AND JOB HISTORY START DATESOF ALL EMPLOYEES.

```
BEGIN
   FOR REC IN (SELECT EMPLOYEE_ID, FIRST_NAME || ' ' || LAST_NAME AS
            NAME, HIRE_DATEFROM EMPLOYEES) LOOP
      DBMS_OUTPUT.PUT_LINE('ID: ' || REC.EMPLOYEE_ID || ', NAME: ' || REC.NAME || ', START
      DATE: '
||
   REC.HIRE_D
   ATE);END
   LOOP;
END;
```

```
ID: 2, Name: Emma Austen, Start Date: 11/06/1990
ID: 10, Name: Paul Rudd, Start Date: 04/06/1969
ID: 11, Name: Brie Zlotkey, Start Date: 10/01/1989
ID: 20, Name: Elizabeth Olsen, Start Date: 02/16/1989
ID: 25, Name: Cate Abu, Start Date: 05/14/1969
ID: 27, Name: Jeff Goldblum, Start Date: 10/22/1952
ID: 122, Name: Robert Downey, Start Date: 04/04/1965
ID: 18, Name: Karen Gillan, Start Date: 11/28/1987
ID: 21, Name: Anthony Mackie, Start Date: 09/23/1978
ID: 22, Name: Sebastian Stan, Start Date: 08/13/1982
ID: 28, Name: Karl Austin, Start Date: 06/07/1972
ID: 176, Name: Chris Morris, Start Date: 05/07/1994
ID: 6, Name: Mark Ruffalo, Start Date: 11/22/1967
ID: 12, Name: Chadwick Boseman, Start Date: 11/29/1976
ID: 24, Name: Tom Hiddleston, Start Date: 02/09/1981
ID: 1, Name: Justin Beiber, Start Date: 09/21/1996
ID: 8, Name: Jeremy Wilson, Start Date: 01/07/1971
ID: 7, Name: Chris Hemsworth, Start Date: 08/11/1983
ID: 9, Name: Tom Holland, Start Date: 06/01/1996
ID: 13, Name: Chris Austin, Start Date: 06/21/1979
ID: 17, Name: Dave Bautista, Start Date: 01/18/1969
ID: 26, Name: Tessa Thompson, Start Date: 10/03/1983
ID: 14, Name: Zoe Austin, Start Date: 06/19/1978
ID: 19, Name: Pom Davies, Start Date: 05/03/1986
ID: 42, Name: Matos roy, Start Date: 02/23/1991
ID: 4, Name: Scarlett Austin, Start Date: 11/22/1984
ID: 15, Name: Bradley Hook, Start Date: 01/05/1975
ID: 16, Name: Vin Diesel, Start Date: 07/18/1967
ID: 110, Name: Benedict andru, Start Date: 07/19/1976
ID: 30, Name: Taika Waititi, Start Date: 08/16/1975
ID: 40, Name: John Doe , Start Date: 08/10/1995
ID: 29, Name: Idris Elba, Start Date: 09/06/1972
231501154@rajalakshmi.edu.in   shriram154   en
Copyright © 1999, 2024, Oracle and/or its affiliates.
```

PROGRAM 15
WRITE A PL/SQL PROGRAM TO DISPLAY THE EMPLOYEE IDS, NAMES, AND JOB HISTORY END DATESOF ALL EMPLOYEES.

```
BEGIN
   FOR REC IN (SELECT EMPLOYEE_ID, FIRST_NAME || ' ' || LAST_NAME AS NAME,
            END_DATEFROM EMPLOYEES)
   LOOP
      DBMS_OUTPUT.PUT_LINE('ID: ' ||
                     REC.EMPLOYEE_ID ||', NAME: ' ||
                     REC.NAME ||
                     ', END DATE: ' ||
                     NVL(TO_CHAR(REC.END_DATE, 'YYYY-MM-DD'), 'STILL ACTIVE'));
   END LOOP;
END;
```

```
ID: 2, Name: Emma Austen, End Date: Still Active
ID: 10, Name: Paul Rudd, End Date: Still Active
ID: 11, Name: Brie Zlotkey, End Date: Still Active
ID: 20, Name: Elizabeth Olsen, End Date: Still Active
ID: 25, Name: Cate Abu, End Date: Still Active
ID: 27, Name: Jeff Goldblum, End Date: Still Active
ID: 122, Name: Robert Downey, End Date: Still Active
ID: 18, Name: Karen Gillan, End Date: Still Active
ID: 21, Name: Anthony Mackie, End Date: Still Active
ID: 22, Name: Sebastian Stan, End Date: Still Active
ID: 28, Name: Karl Austin, End Date: Still Active
ID: 176, Name: Chris Morris, End Date: Still Active
ID: 6, Name: Mark Ruffalo, End Date: Still Active
ID: 12, Name: Chadwick Boseman, End Date: Still Active
ID: 24, Name: Tom Hiddleston, End Date: Still Active
ID: 1, Name: Justin Beiber, End Date: Still Active
ID: 8, Name: Jeremy Wilson, End Date: Still Active
ID: 7, Name: Chris Hemsworth, End Date: Still Active
ID: 9, Name: Tom Holland, End Date: Still Active
ID: 13, Name: Chris Austin, End Date: Still Active
ID: 17, Name: Dave Bautista, End Date: Still Active
ID: 26, Name: Tessa Thompson, End Date: Still Active
ID: 14, Name: Zoe Austin, End Date: Still Active
ID: 19, Name: Pom Davies, End Date: Still Active
ID: 42, Name: Matos roy, End Date: Still Active
ID: 4, Name: Scarlett Austin, End Date: Still Active
ID: 15, Name: Bradley Hook, End Date: Still Active
ID: 16, Name: Vin Diesel, End Date: Still Active
ID: 110, Name: Benedict andru, End Date: Still Active
ID: 30, Name: Taika Waititi, End Date: Still Active
ID: 40, Name: John Doe , End Date: Still Active
ID: 29, Name: Idris Elba, End Date: Still Active
231501154@rajalakshmi.edu.in   shriram154   en
Copyright © 1999, 2024, Oracle and/or its affiliates.
```