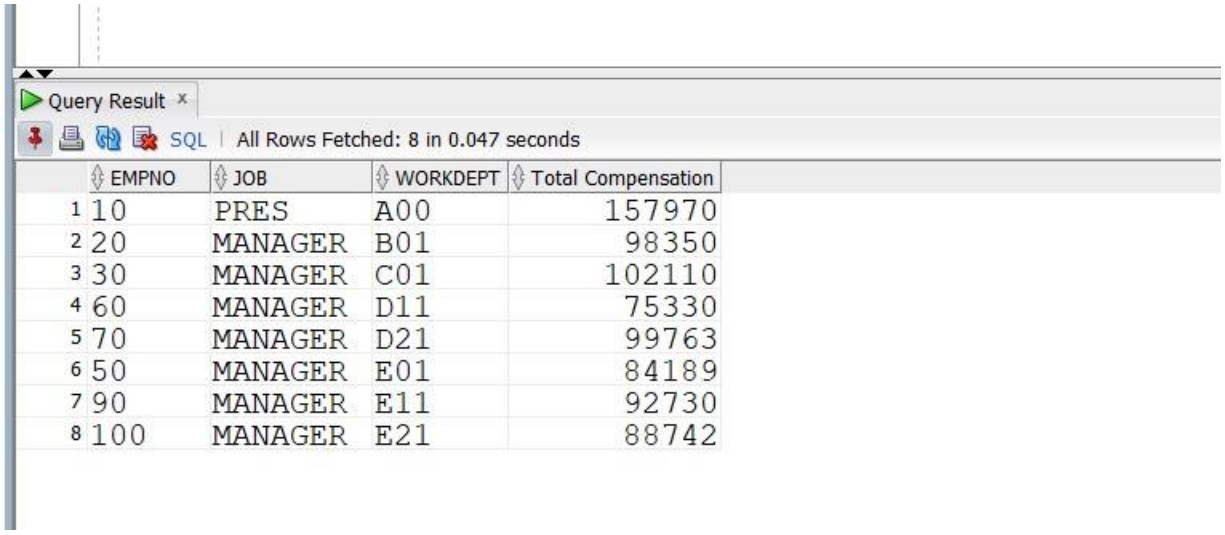
Question 1: Write a query which lists the employee (from EMPLOYEE table) with the highest total compensation (includes SALARY, BONUS and COMMISSION) by department and job type

OUTPUT:


Query Result

SQL | All Rows Fetched: 8 in 0.047 seconds

	EMPNO	JOB	WORKDEPT	Total Compensation
1	10	PRES	A00	157970
2	20	MANAGER	B01	98350
3	30	MANAGER	C01	102110
4	60	MANAGER	D11	75330
5	70	MANAGER	D21	99763
6	50	MANAGER	E01	84189
7	90	MANAGER	E11	92730
8	100	MANAGER	E21	88742

Question 2: Write a query which shows the complete list of last names from both the EMPLOYEE table and STAFF table. Make sure your query is case insensitive (ie SMITH = Smith = smith).

OUTPUT:

LASTNAME
Abrahams
Adamson
Alonzo
Brown
Burke
Daniels
Davis
Edwards
Fraye
Gafney
Geyer
Gonzales
Gounot
Graham
Haas
Hanes
Hemminger
Henderson
James
Jefferson
John
Johnson
Jones
Kermisch
Koonitz
Kwan
Lea
Lee
Lu
Lucchessi
Lundquist
Lutz
Marenghi
Marino
Mehta
Molinare
Monteverde
Natz

LASTNAME
Naughton
Ngan
Nicholls
O'Brien
O'Connell
Orlando
Parker
Perez
Pernal
Pianka
Plotz
Pulaski
Quigley
Quill
Quintana
Rothman
Sanders
Schneider
Schwartz
Scoutten
Setright
Smith
Sneider
Spenser
Springer
Stern
Thompson
Walker
Wheeler
Williams
Wilson
Wong
Yamaguchi
Yamamoto
Yoshimura

Question 3: Write a query which shows where we have two employees assigned to the same employee number, when looking across both EMPLOYEE table and STAFF table.

OUTPUT:

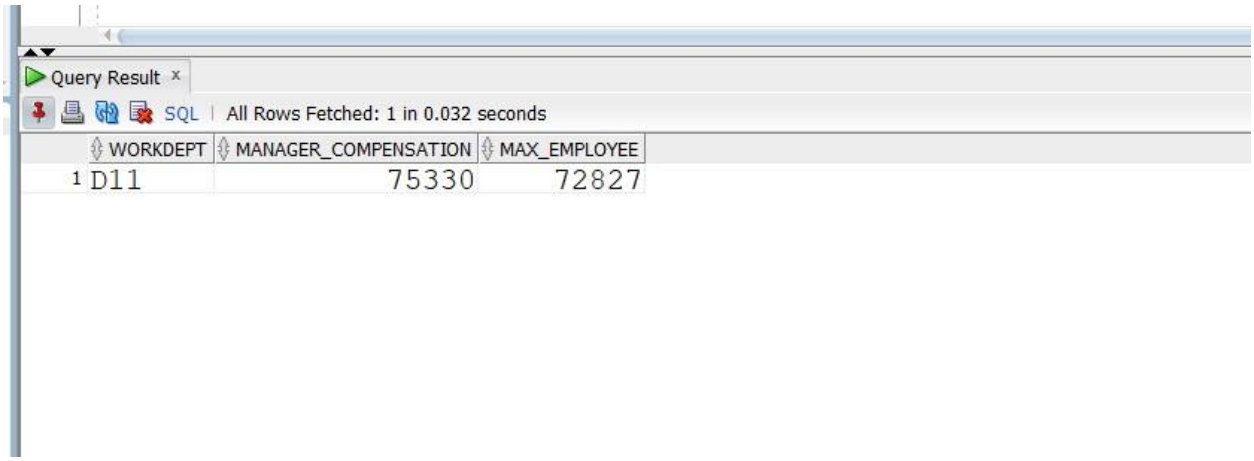
EMPNO	EMPLOYEE LASTNAME	STAFF LASTNAME
10	HAAS	Sanders
100	SPENSER	Plotz
110	LUCCHESSI	Ngan
120	O'CONNELL	Naughton
130	QUINTANA	Yamaguchi
140	NICHOLLS	Fraye
150	ADAMSON	Williams
160	PIANKA	Molinare
170	YOSHIMURA	Kermisch
180	SCOUTTEN	Abrahams
190	WALKER	Sneider
20	THOMPSON	Pernal
200	BROWN	Scoutten
210	JONES	Lu
220	LUTZ	Smith
230	JEFFERSON	Lundquist
240	MARINO	Daniels
250	SMITH	Wheeler
260	JOHNSON	Jones
270	PEREZ	Lea
280	SCHNEIDER	Wilson
290	PARKER	Quill
30	KWAN	Marenghi
300	SMITH	Davis
310	SETRIGHT	Graham
320	MEHTA	Gonzales
330	LEE	Burke
340	GOUNOT	Edwards
50	GEYER	Hanes
60	STERN	Quigley
70	PULASKI	Rothman
90	HENDERSON	Koonitz

Question 4: Write a query which lists all employees across both the STAFF and EMPLOYEE table, which have an 'oo' OR a 'z' in their last name.

OUTPUT:

Question 5: Write a query which looks at the EMPLOYEE table and, for each department, compares the manager's total compensation (SALARY, BONUS and COMMISSION) to the top paid employee's total compensation and displays output if the top paid employee in that department makes within $10,000 in total compensation as compared to their manager

OUTPUT:



Question 6: Write a query which looks across both the EMPLOYEE and STAFF table and returns the total "variable pay" (COMMISSION + BONUS) for each employee.

OUTPUT:

LASTNAME	VARIABLE_PAY
Abrahams	236.5
Adamson	2522
Alonzo	2407
Brown	2817
Burke	55.5
Daniels	0
Davis	806.1
Edwards	1285
Fraye	0
Gafney	188
Geyer	4014
Gonzales	844
Gounot	2407
Graham	200.3
Haas	5220
Hanes	0
Hemminger	5220
Henderson	2980
James	128.2
Jefferson	2174
John	2987
Johnson	1680
Jones	0
Jones	1862
Kermisch	110.1
Koonitz	1386.7
Kwan	3860
Lea	0
Lee	2530
Lu	0
Lucchessi	4620
Lundquist	189.65
Lutz	2987
Marenghi	0
Marino	2901
Mehta	1996
Molinare	0
Monteverde	2901

LASTNAME	VARIABLE_PAY
Natz	2874
Naughton	180
Ngan	206.6
Nicholls	2874
O'Brien	846.55
O'Connell	2940
Orlando	2940
Parker	1527
Perez	2690
Pernal	612.45
Pianka	2180
Plotz	0
Pulaski	3593
Quigley	650.25
Quill	0
Quintana	2404
Rothman	1152
Sanders	0
Schneider	2600
Schwartz	2600
Scoutten	84.2
Scoutten	2207
Setright	1572
Smith	992.8
Smith	1820
Smith	1934
Sneider	126.5
Spenser	2592
Springer	1572
Stern	3080
Thompson	4100
Walker	2036
Wheeler	513.3
Williams	637.65
Wilson	811.5
Wong	2530
Yamaguchi	75.6
Yamamoto	2474

LASTNAME	VARIABLE_PAY
Yoshimura	2474

Question 7: Write a stored procedure for the EMPLOYEE table which takes, as input, an employee number, and a rating of either 1, 2 or 3.

OUTPUT:

Procedure UPDATE_SALARY compiled

VALID ID WITH RATING 1

EMP ID: 10

PREV SALARY: 152750

PREV BONUS: 1000

PREV COMM: 4220

NEW SALARY: 162750

NEW BONUS: 1300

NEW COMM: 4431

VALID ID WITH RATING 2

EMP ID: 20

PREV SALARY: 94250

PREV BONUS: 800

PREV COMM: 3300

NEW SALARY: 99250

NEW BONUS: 1000

NEW COMM: 3366

VALID ID WITH RATING 3

EMP ID: 30

PREV SALARY: 98250

PREV BONUS: 800

PREV COMM: 3060

NEW SALARY: 100250

NEW BONUS: 800

NEW COMM: 3060

INVALID ID WITH RATING 1

ERROR: INVALID Employee ID

VALID ID WITH INVALID RATING OF 4

ERROR: INVALID RATING

PL/SQL procedure successfully completed.

Question 8: Write a stored procedure for the EMPLOYEE table which takes employee number and education level upgrade as input - and - increases the education level of the employee based on the input.

OUTPUT:

Procedure UPDATE_EDUCATION compiled

UPDATING JOHN TO LEVEL 16

EMPLOYEE ID: 50

PREV EDUCATION LEVEL: 16

NEW EDUCATION LEVEL: 16

UPDATING JOHN TO LEVEL 19

EMPLOYEE ID: 50

PREV EDUCATION LEVEL: 16

NEW EDUCATION LEVEL: 19

UPDATING JOHN TO LEVEL 20

EMPLOYEE ID: 50

PREV EDUCATION LEVEL: 16

NEW EDUCATION LEVEL: 20

UPDATING JOHN TO LEVEL 23

EMPLOYEE ID: 50

PREV EDUCATION LEVEL: 16

NEW EDUCATION LEVEL: 23

UPDATING JOHN TO LEVEL 25

EMPLOYEE ID: 50

PREV EDUCATION LEVEL: 16

NEW EDUCATION LEVEL: 25

UPDATING SALLY TO LEVEL 16. IT SHOULD GIVE ERROR

ERROR: CANNOT REDUCE EXISTING EDUCATION LEVEL

UPDATING SALLY TO LEVEL 19. IT SHOULD GIVE ERROR

ERROR: CANNOT REDUCE EXISTING EDUCATION LEVEL

UPDATING WITH INVALID ID. IT SHOULD GIVE ERROR

ERROR: EMPLOYEE ID

PL/SQL procedure successfully completed.

Question 9: Write a function called PHONE which takes an employee number as input and displays a full phone number for that employee, using the PHONENO value as part of the function.

Question 10: Write a stored procedure which calls your PHONE function.

OUTPUT of 9 and 10:

Table EMPLOYEE altered.

Function PHONE compiled

```
CREATE OR REPLACE PROCEDURE updatephone
AS
   empid employee.empno%TYPE;
   dept employee.workdept%TYPE;
   name employee.firstname%TYPE;
   phone employee.phoneno%TYPE;
   phonenum employee.phoneno%TYPE;
CURSOR c1 IS

SELECT
   empno, workdept, firstname , phoneno, phone(empno)
FROM employee emp
WHERE workdept like 'E%';

BEGIN

   OPEN c1;

   LOOP
      FETCH c1
      INTO
         empid, dept, name, phone, phonenum;
```

```
    EXIT WHEN c1%NOTFOUND;
    UPDATE employee
      SET phonenum  = phonenum
      WHERE empno = empid;
    DBMS_OUTPUT.PUT_LINE(dept || ' '||name||' ' || phone || ' ' || phonenum);
    END LOOP;
    CLOSE c1;

END;
```

Error report -

ORA-06550: line 10, column 1:

PLS-00103: Encountered the symbol "CREATE"

06550. 00000 - "line %s, column %s:\n%s"

*Cause: Usually a PL/SQL compilation error.

*Action:

PHONE NUMBER: (416) 123-6789

E01 JOHN 6789 (416) 123-6789

PHONE NUMBER: (416) 123-5498

E11 EILEEN 5498 (416) 123-5498

PHONE NUMBER: (416) 123-0972

E21 THEODORE 0972 (416) 123-0972

PHONE NUMBER: (416) 123-8997

E11 ETHEL 8997 (416) 123-8997

PHONE NUMBER: (416) 123-4502

E11 JOHN 4502 (416) 123-4502

PHONE NUMBER: (416) 123-2095

E11 PHILIP 2095 (416) 123-2095

PHONE NUMBER: (416) 123-3332

E11 MAUDE 3332 (416) 123-3332

PHONE NUMBER: (416) 123-9990

E21 RAMLAL 9990 (416) 123-9990

PHONE NUMBER: (416) 123-2103

E21 WING 2103 (416) 123-2103

PHONE NUMBER: (416) 123-5698

E21 JASON 5698 (416) 123-5698

PHONE NUMBER: (416) 123-8997

E11 EILEEN 8997 (416) 123-8997

PHONE NUMBER: (416) 123-3332

E11 MICHELLE 3332 (416) 123-3332

PHONE NUMBER: (416) 123-2103

E21 HELENA 2103 (416) 123-2103

PHONE NUMBER: (416) 123-5698

E21 ROY 5698 (416) 123-5698

PL/SQL procedure successfully completed.