QUEZON CITY UNIVERSITY

COLLEGE OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

IM101 – Advanced Database Systems

Week-5-6 Lab Activity – SQL Review

NAME: Jose, Rommel O.

STUDENT NO: 23-2300

YEAR/SECTION: 2nd Yr. SBIT-2A

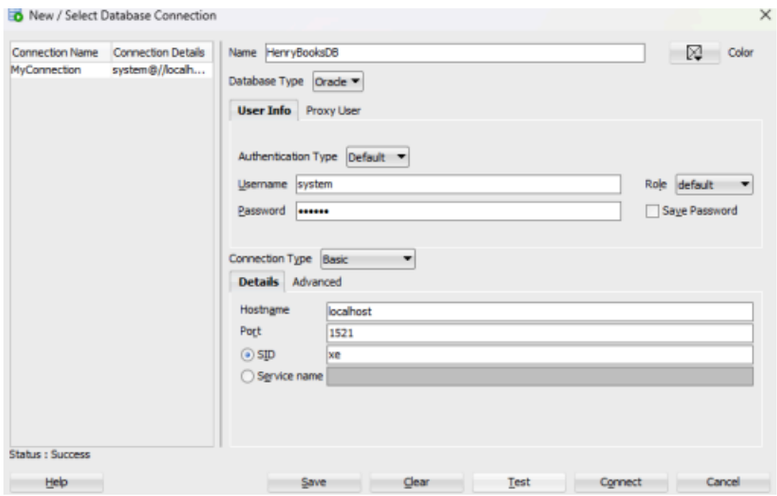
DATE: February 16, 2025

SCORE	PERCENTAGE

Instruction.

- Provide screenshots that shows your Oracle account, code, and output.
- Screenshots must not be whole screen.
- Text must be readable.

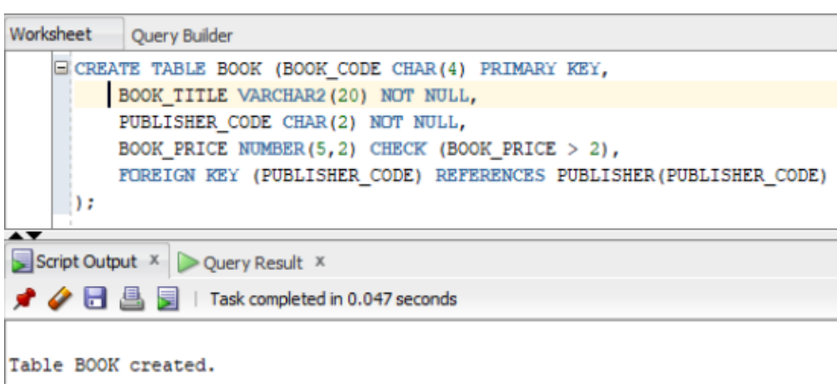
Part-1. Henry Books Database.

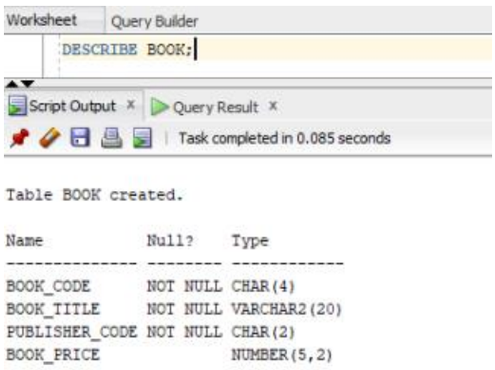


1. Create the following tables and all its constraints.

BOOK

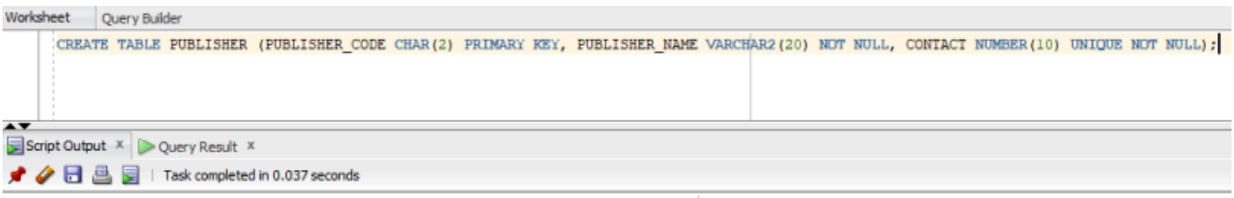
Column	Type	Length	Decimal Places	Nulls Allowed?	Description
BOOK_CODE	CHAR	4		NO	Primary key
BOOK_TITLE	VARCHAR	20		NO	Book title
PUBLISHER_CODE	CHAR	2		NO	Foreign key to Publisher table
BOOK_PRICE	NUMBER	5	2	NO	Value must be greater than 2.

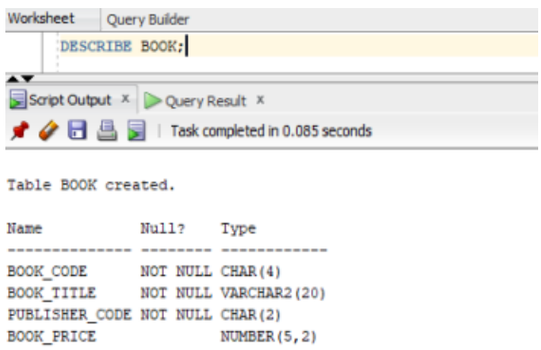




PUBLISHER

Column	Type	Length	Decimal Places	Nulls Allowed?	Description
PUBLISHER_CODE	CHAR	2		NO	Primary key
PUBLISHER_NAME	VARCHAR	20		NO	Publisher name
CONTACT	NUMERIC	10		NO	Must be unique

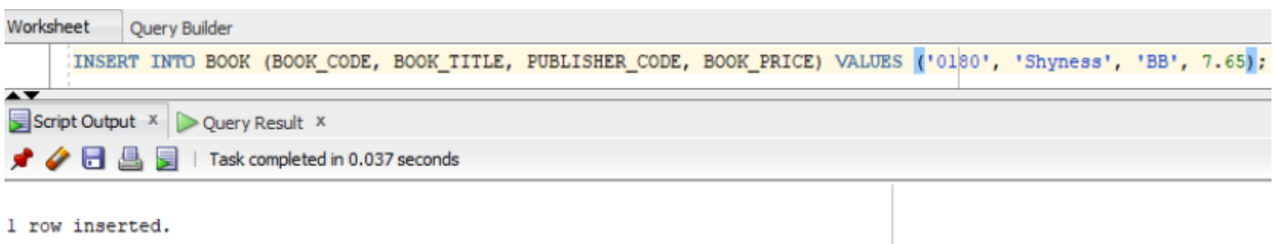




2. Insert the following records.

BOOK

BOOK_CODE	BOOK_TITLE	PUBLISHER_CODE	BOOK_PRICE
0180	Shyness	BB	7.65
0189	Kane and Abel	PB	5.55
0378	Dunwich Horror and Others	PB	19.75
0808	Knockdown	PB	6.50



QUEZON CITY UNIVERSITY

COLLEGE OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

```
INSERT INTO BOOK (BOOK_CODE, BOOK_TITLE, PUBLISHER_CODE, BOOK_PRICE) VALUES ('0189', 'Kane and Abel', 'PB', 5.55);
```

Script Output | Query Result | Task completed in 0.033 seconds

```
1 row inserted.
```

```
INSERT INTO BOOK (BOOK_CODE, BOOK_TITLE, PUBLISHER_CODE, BOOK_PRICE) VALUES ('0378', 'Dunwich Horror and Others', 'PB', 19.75);
```

Script Output | Query Result | Task completed in 0.034 seconds

```
1 row inserted.
```

Worksheet | Query Builder

```
INSERT INTO BOOK (BOOK_CODE, BOOK_TITLE, PUBLISHER_CODE, BOOK_PRICE) VALUES ('0808', 'Knockdown', 'PB', 6.50);
```

Script Output | Query Result | Task completed in 0.035 seconds

```
1 row inserted.
```

Worksheet | Query Builder

```
SELECT * FROM BOOK;
```

Script Output | Query Result | SQL | All Rows Fetched: 4 in 0.003 seconds

	BOOK_CODE	BOOK_TITLE	PUBLISHER_CODE	BOOK_PRICE
1	0180	Shyness	BB	7.65
2	0189	Kane and Abel	PB	5.55
3	0378	Dunwich Horror and Others	PB	19.75
4	0808	Knockdown	PB	6.5

PUBLISHER

PUBLISHER_CODE	DESCRIPTION	CONTACT
BB	Bantam Books	9391234560
PB	Pocket Books	9362345678
SI	Signet	9352345685

Worksheet | Query Builder

```
INSERT INTO PUBLISHER (PUBLISHER_CODE, PUBLISHER_NAME, CONTACT) VALUES
('BB', 'Bantam Books', 9391234560);
```

Script Output | Query Result | Task completed in 0.039 seconds

```
1 row inserted.
```

Worksheet | Query Builder

```
INSERT INTO PUBLISHER (PUBLISHER_CODE, PUBLISHER_NAME, CONTACT) VALUES
('PB', 'Pocket Books', 9362345678);
```

Script Output | Query Result | Task completed in 0.033 seconds

```
1 row inserted.
```

Worksheet | Query Builder

```
INSERT INTO PUBLISHER (PUBLISHER_CODE, PUBLISHER_NAME, CONTACT) VALUES
('SI', 'Signet', 9352345685);
```

Script Output | Query Result | Task completed in 0.034 seconds

```
1 row inserted.
```

QUEZON CITY UNIVERSITY

COLLEGE OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

Worksheet | Query Builder

```
SELECT * FROM PUBLISHER;
```

Script Output | Query Result

SQL | All Rows Fetched: 3 in 0.004 seconds

	PUBLISHER_CODE	PUBLISHER_NAME	CONTACT
1	BB	Bantam Books	9391234560
2	PB	Pocket Books	9362345678
3	SI	Signet	9352345685

Part-2. Use the HR Schema and create the appropriate SQL statement for the following queries.

1. The HR department needs a report that displays the last name and salary of employees who earn more than 10,000 from the Marketing, Purchasing, and Accounting Department.

Worksheet | Query Builder

```
SELECT last_name, salary
FROM employees
WHERE salary > 10000
AND department_id IN (
    SELECT department_id
    FROM departments
    WHERE department_name IN ('Marketing', 'Purchasing', 'Accounting')
);
```

Query Result

SQL | All Rows Fetched: 3 in 0.005 seconds

	LAST_NAME	SALARY
1	Hartstein	13000
2	Raphaely	11000
3	Higgins	12008

2. Display the employee id, last name, salary as 'Old Salary', salary increased by 10% as 'New Salary', another column that displays the new salary less the old salary as 'Increase'. Include in the list only those employees whose salary is not in the range of $5,000 to $12,000. Display all numeric values with two decimal places. Sort the list in descending order of New Salary.

Worksheet | Query Builder

```
SELECT employee_id,
       last_name,
       TO_CHAR(salary, '999,999.00') AS "Old Salary",
       TO_CHAR(salary * 1.10, '999,999.00') AS "New Salary",
       TO_CHAR((salary * 1.10) - salary, '999,999.00') AS "Increase"
FROM employees
WHERE salary NOT BETWEEN 5000 AND 12000
ORDER BY salary * 1.10 DESC;
```

Query Result

SQL | All Rows Fetched: 57 in 0.008 seconds

	EMPLOYEE_ID	LAST_NAME	Old Salary	New Salary	Increase
1	100	King	24,000.00	26,400.00	2,400.00
2	101	Kochhar	17,000.00	18,700.00	1,700.00
3	102	De Haan	17,000.00	18,700.00	1,700.00
4	145	Russell	14,000.00	15,400.00	1,400.00
5	146	Partners	13,500.00	14,850.00	1,350.00
6	201	Hartstein	13,000.00	14,300.00	1,300.00
7	108	Greenberg	12,008.00	13,208.80	1,200.80
8	205	Higgins	12,008.00	13,208.80	1,200.80
9	105	Austin	4,800.00	5,280.00	480.00
10	106	Pataballa	4,800.00	5,280.00	480.00
11	200	Whalen	4,400.00	4,840.00	440.00
12	107	Lorentz	4,200.00	4,620.00	420.00
13	184	Sarchand	4,200.00	4,620.00	420.00
14	185	Bull	4,100.00	4,510.00	410.00
15	192	Bell	4,000.00	4,400.00	400.00
16	193	Everett	3,900.00	4,290.00	390.00
17	188	Chung	3,800.00	4,180.00	380.00
18	189	Dilly	3,600.00	3,960.00	360.00
19	137	Ladwig	3,600.00	3,960.00	360.00
20	141	Rajs	3,500.00	3,850.00	350.00
21	186	Dellinger	3,400.00	3,740.00	340.00
22	133	Mallin	3,300.00	3,630.00	330.00
23	129	Bissot	3,300.00	3,630.00	330.00
24	125	Nayer	3,200.00	3,520.00	320.00
25	194	McCain	3,200.00	3,520.00	320.00
26	138	Stiles	3,200.00	3,520.00	320.00
27	180	Taylor	3,200.00	3,520.00	320.00
28	115	Khoo	3,100.00	3,410.00	310.00
29	196	Walsh	3,100.00	3,410.00	310.00
30	181	Fleaur	3,100.00	3,410.00	310.00
31	142	Davies	3,100.00	3,410.00	310.00
32	197	Feeney	3,000.00	3,300.00	300.00
33	187	Cabrio	3,000.00	3,300.00	300.00
34	116	Baida	2,900.00	3,190.00	290.00
35	190	Gates	2,900.00	3,190.00	290.00
36	134	Rogers	2,900.00	3,190.00	290.00
37	183	Geoni	2,800.00	3,080.00	280.00
38	195	Jones	2,800.00	3,080.00	280.00
39	117	Tobias	2,800.00	3,080.00	280.00
40	130	Atkinson	2,800.00	3,080.00	280.00
41	126	Mikkilineni	2,700.00	2,970.00	270.00
42	139	Seo	2,700.00	2,970.00	270.00
43	198	OConnell	2,600.00	2,860.00	260.00
44	199	Grant	2,600.00	2,860.00	260.00
45	118	Himuro	2,600.00	2,860.00	260.00
46	143	Matos	2,600.00	2,860.00	260.00
47	144	Vargas	2,500.00	2,750.00	250.00
48	131	Marlow	2,500.00	2,750.00	250.00
49	119	Colmenares	2,500.00	2,750.00	250.00

49	119	Colmenares	2,500.00	2,750.00	250.00
50	182	Sullivan	2,500.00	2,750.00	250.00
51	140	Patel	2,500.00	2,750.00	250.00
52	191	Perkins	2,500.00	2,750.00	250.00
53	135	Gee	2,400.00	2,640.00	240.00
54	127	Landry	2,400.00	2,640.00	240.00
55	136	Philtanker	2,200.00	2,420.00	220.00
56	128	Markle	2,200.00	2,420.00	220.00
57	132	Olson	2,100.00	2,310.00	210.00

3. Create a report to display the last name, job ID, and hire date for employees with the last names of Matos and Taylor. Order the query in ascending order by the hire date.

```
SELECT last_name, job_id, hire_date
FROM employees
WHERE last_name IN ('Matos', 'Taylor')
ORDER BY hire_date ASC;
```

Query Result

SQL | All Rows Fetched: 3 in 0.005 seconds

	LAST_NAME	JOB_ID	HIRE_DATE
1	Taylor	SH_CLERK	24-JAN-06
2	Matos	ST_CLERK	15-MAR-06
3	Taylor	SA_REP	24-MAR-06

4. The HR department wants to find the duration of employment for each employee. For each employee, display the last name and calculate the number of months between today and the date on which the employees was hired. Label the column s MONTHS_WORKED. Order your results by the number of months employed. Round the number of months up to the closest whole number.

Worksheet | Query Builder

```
SELECT last_name,
       CEIL(MONTHS_BETWEEN(SYSDATE, hire_date)) AS MONTHS_WORKED
FROM employees
ORDER BY MONTHS_WORKED;
```

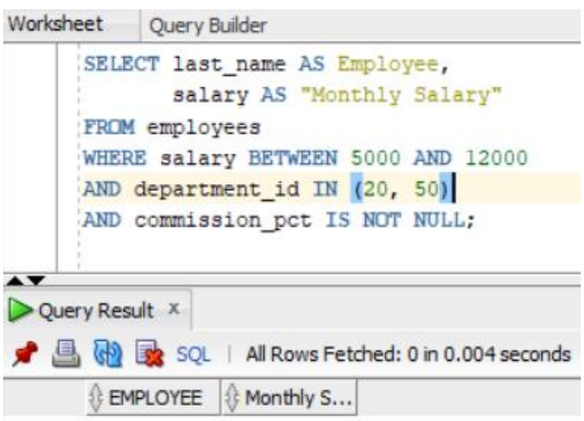
Query Result

SQL | Fetched 50 rows in 0.016 seconds

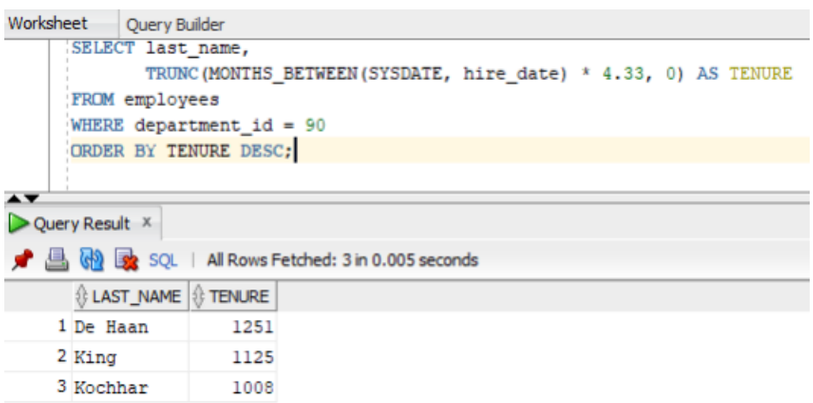
	LAST_NAME	MONTHS_WORKED
1	Banda	202
2	Kumar	202
3	Ande	203
4	Markle	204
5	Lee	204
6	Philtanker	205
7	Zlotkey	205
8	Marvins	205
9	Geoni	205
10	Johnson	206
11	Perkins	206
12	Grant	206
13	Popp	207
14	Mourgos	207
15	Gee	207
16	Tuvault	207
17	Cambrault	208
18	Colmenares	211
19	Sullivan	212
20	OConnell	212
21	Ernst	213
22	Grant	213
23	Olson	215
24	Greene	215
25	Bates	215
26	Jones	215
27	Smith	216
28	Lorentz	217
29	Cabrio	217
30	Landry	218
31	Himuro	219
32	Cambrault	219
33	Sewall	220
34	Mikkilineni	221
35	Rogers	222
36	Vargas	224
37	Dellinger	224
38	Gates	224
39	McCain	224
40	Feeney	225
41	Livingston	226
42	Walsh	226
43	Patel	227
44	Matos	227
45	Olsen	227
46	Bloom	227
47	Taylor	227
48	Urman	228
49	Fleaur	228
50	Pataballa	229
51	Seo	229
52	Fox	229
53	Taylor	229
54	Hunold	230
55	Baida	230

56	Doran	230
57	Atkinson	232
58	Stiles	232
59	Vishney	232
60	Kochhar	233
61	Chen	233
62	Sciarra	233
63	Vollman	233
64	Bissot	234
65	Hall	234
66	Fay	234
67	Tobias	235
68	Nayer	235
69	Dilly	235
70	Austin	236
71	Chung	237
72	Fripp	239
73	Bernstein	239
74	Hutton	239
75	Marlow	240
76	Errazuriz	240
77	Smith	240
78	Ozer	240
79	Bull	240
80	Everett	240
81	Davies	241
82	Tucker	241
83	Partners	242
84	Russell	245
85	Weiss	247
86	McEwen	247
87	Mallin	249
88	Abel	250
89	Sully	252
90	Hartstein	252
91	King	253
92	Sarchand	253
93	Bell	253
94	Rajs	256
95	Whalen	257
96	King	260
97	Ladwig	260
98	Khoo	261
99	Kaufling	262
100	Raphaely	267
101	Faviet	270
102	Greenberg	270
103	Gietz	273
104	Mavris	273
105	Baer	273
106	Higgins	273
107	De Haan	290

5. Display the last name and salary of employees who earn between 5,000 and 12,000 and are in department 20 or 50. Label the columns Employee and Monthly Salary, respectively. Include only those employees with commission.



6. Create a query to display the last name and the number of weeks employed for all employees in department 90. Label the number of weeks column as TENURE. Truncate the number of weeks value to 0 decimal places. Show the records in descending order of the employees' tenure.



7. Create a report to display the last name, salary, and commission of all employees who earn commissions. Sort data in descending order of salary and commissions. Use the column's numeric position in the ORDER BY clause.

```
Worksheet | Query Builder
SELECT last_name, salary, commission_pct
FROM employees
WHERE commission_pct IS NOT NULL
ORDER BY 2 DESC, 3 DESC;
```

Query Result — All Rows Fetched: 35 in 0.008 seconds

	LAST_NAME	SALARY	COMMISSION_PCT
1	Russell	14000	0.4
2	Partners	13500	0.3
3	Errazuriz	12000	0.3
4	Ozer	11500	0.25
5	Cambrault	11000	0.3
6	Abel	11000	0.3
7	Vishney	10500	0.25
8	Zlotkey	10500	0.2
9	King	10000	0.35
10	Tucker	10000	0.3
11	Bloom	10000	0.2
12	Fox	9600	0.2
13	Sully	9500	0.35
14	Bernstein	9500	0.25
15	Greene	9500	0.15
16	McEwen	9000	0.35
17	Hall	9000	0.25
18	Hutton	8800	0.25
19	Taylor	8600	0.2
20	Livingston	8400	0.2
21	Smith	8000	0.3
22	Olsen	8000	0.2
23	Doran	7500	0.3
24	Cambrault	7500	0.2
25	Smith	7400	0.15
26	Bates	7300	0.15
27	Marvins	7200	0.1
28	Sewall	7000	0.25
29	Tuvault	7000	0.15
30	Grant	7000	0.15
31	Lee	6800	0.1
32	Ande	6400	0.1
33	Banda	6200	0.1
34	Johnson	6200	0.1
35	Kumar	6100	0.1

8. Find the highest, lowest, sum, and average salary of all employees for each job type. Label the columns as Maximum, Minimum, Sum, and Average, respectively. Round your results to the nearest whole number.

```
Worksheet | Query Builder
SELECT job_id,
       ROUND(MAX(salary)) AS Maximum,
       ROUND(MIN(salary)) AS Minimum,
       ROUND(SUM(salary)) AS Sum,
       ROUND(AVG(salary)) AS Average
FROM employees
GROUP BY job_id;
```

Query Result — All Rows Fetched: 19 in 0.007 seconds

	JOB_ID	MAXIMUM	MINIMUM	SUM	AVERAGE
1	IT_PROG	9000	4200	28800	5760
2	AC_MGR	12008	12008	12008	12008
3	AC_ACCOUNT	8300	8300	8300	8300
4	ST_MAN	8200	5800	36400	7280
5	PU_MAN	11000	11000	11000	11000
6	AD_ASST	4400	4400	4400	4400
7	AD_VP	17000	17000	34000	17000
8	SH_CLERK	4200	2500	64300	3215
9	FI_ACCOUNT	9000	6900	39600	7920
10	FI_MGR	12008	12008	12008	12008
11	PU_CLERK	3100	2500	13900	2780
12	SA_MAN	14000	10500	61000	12200
13	MK_MAN	13000	13000	13000	13000
14	PR_REP	10000	10000	10000	10000
15	AD_PRES	24000	24000	24000	24000
16	SA_REP	11500	6100	250500	8350
17	MK_REP	6000	6000	6000	6000
18	ST_CLERK	3600	2100	55700	2785
19	HR_REP	6500	6500	6500	6500

9. How many employees are there for each department? Order the list by department id in ascending order.

```
SELECT department_id, COUNT(*) AS Employee_Count
FROM employees
GROUP BY department_id
ORDER BY department_id ASC;
```

Query Result | SQL | All Rows Fetched: 12 in 0.004 seconds

	DEPARTMENT_ID	EMPLOYEE_COUNT
1	10	1
2	20	2
3	30	6
4	40	1
5	50	45
6	60	5
7	70	1
8	80	34
9	90	3
10	100	6
11	110	2
12	(null)	1

10. Create a report to display the manager number and salary of the lowest-paid employee for the manager. Exclude anyone whose manager is not known. Exclude any groups where the minimum salary is $6,000 or less. Sort the output in descending order of salary.

Worksheet | Query Builder

```
SELECT manager_id, MIN(salary) AS Lowest_Salary
FROM employees
WHERE manager_id IS NOT NULL
GROUP BY manager_id
HAVING MIN(salary) > 6000
ORDER BY Lowest_Salary DESC;
```

Query Result | SQL | All Rows Fetched: 8 in 0.006 seconds

	MANAGER_ID	LOWEST_SALARY
1	102	9000
2	205	8300
3	145	7000
4	146	7000
5	108	6900
6	147	6200
7	149	6200
8	148	6100

Part-3. Use the HR Schema and create the appropriate SQL statement for the following queries.

11. Create a query to display the total number of employees and, of that total, the number of employees hired in 1995, 1996, 1997, and 1998. Create appropriate headings.

Worksheet | Query Builder

```
SELECT COUNT(*) AS "Total Employees",
       SUM(CASE WHEN EXTRACT(YEAR FROM hire_date) = 1995 THEN 1 ELSE 0 END) AS "Hired in 1995",
       SUM(CASE WHEN EXTRACT(YEAR FROM hire_date) = 1996 THEN 1 ELSE 0 END) AS "Hired in 1996",
       SUM(CASE WHEN EXTRACT(YEAR FROM hire_date) = 1997 THEN 1 ELSE 0 END) AS "Hired in 1997",
       SUM(CASE WHEN EXTRACT(YEAR FROM hire_date) = 1998 THEN 1 ELSE 0 END) AS "Hired in 1998"
FROM employees;
```

Query Result | SQL | All Rows Fetched: 1 in 0.007 seconds

	Total Employees	Hired in 1995	Hired in 1996	Hired in 1997	Hired in 1998
1	107	0	0	0	0

QUEZON CITY UNIVERSITY

COLLEGE OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

12. Display the last name, job, and salary for all employees whose jobs are either those of a sales representative or of a stock clerk, and whose salaries are not equal to $2,500, $3,500, or $7,000.

```
SELECT last_name, job_id, salary
FROM employees
WHERE job_id IN ('SA_REP', 'ST_CLERK')
AND salary NOT IN (2500, 3500, 7000);
```

Query Result | SQL | All Rows Fetched: 43 in 0.004 seconc

	LAST_NAME	JOB_ID	SALARY
1	Nayer	ST_CLERK	3200
2	Mikkilineni	ST_CLERK	2700
3	Landry	ST_CLERK	2400
4	Markle	ST_CLERK	2200
5	Bissot	ST_CLERK	3300
6	Atkinson	ST_CLERK	2800
7	Olson	ST_CLERK	2100
8	Mallin	ST_CLERK	3300
9	Rogers	ST_CLERK	2900
10	Gee	ST_CLERK	2400
11	Philtanker	ST_CLERK	2200
12	Ladwig	ST_CLERK	3600
13	Stiles	ST_CLERK	3200
14	Seo	ST_CLERK	2700
15	Davies	ST_CLERK	3100
16	Matos	ST_CLERK	2600
17	Tucker	SA_REP	10000
18	Bernstein	SA_REP	9500
19	Hall	SA_REP	9000
20	Olsen	SA_REP	8000
21	Cambrault	SA_REP	7500
22	King	SA_REP	10000
23	Sully	SA_REP	9500
24	McEwen	SA_REP	9000
25	Smith	SA_REP	8000
26	Doran	SA_REP	7500
27	Vishney	SA_REP	10500
28	Greene	SA_REP	9500
29	Marvins	SA_REP	7200
30	Lee	SA_REP	6800
31	Ande	SA_REP	6400
32	Banda	SA_REP	6200
33	Ozer	SA_REP	11500
34	Bloom	SA_REP	10000
35	Fox	SA_REP	9600
36	Smith	SA_REP	7400
37	Bates	SA_REP	7300
38	Kumar	SA_REP	6100
39	Abel	SA_REP	11000
40	Hutton	SA_REP	8800
41	Taylor	SA_REP	8600
42	Livingston	SA_REP	8400
43	Johnson	SA_REP	6200

13. Using the DECODE function, write a query that displays the grade of all employees based on the value of the column JOB_ID, using the following data:

Job	Grade
AD_PRES	A
ST_MAN	B
IT_PROG	C
SA_REP	D
ST_CLERK	E
None of the above	0

```
SELECT last_name, job_id,
       DECODE(job_id,
              'AD_PRES', 'A',
              'ST_MAN', 'B',
              'IT_PROG', 'C',
              'SA_REP', 'D',
              'ST_CLERK', 'E',
              '0') AS Grade
FROM employees;
```

Query Result | SQL | Fetched 100 rows in 0.00

	LAST_NAME	JOB_ID	GRADE
1	Abel	SA_REP	D
2	Ande	SA_REP	D
3	Atkinson	ST_CLERK	E
4	Austin	IT_PROG	C
5	Baer	PR_REP	0
6	Baida	PU_CLERK	0
7	Banda	SA_REP	D
8	Bates	SA_REP	D
9	Bell	SH_CLERK	0
10	Bernstein	SA_REP	D
11	Bissot	ST_CLERK	E
12	Bloom	SA_REP	D
13	Bull	SH_CLERK	0
14	Cabrio	SH_CLERK	0
15	Cambrault	SA_MAN	0
16	Cambrault	SA_REP	D
17	Chen	FI_ACCOUNT	0
18	Chung	SH_CLERK	0
19	Colmenares	PU_CLERK	0
20	Davies	ST_CLERK	E
21	De Haan	AD_VP	0
22	Dellinger	SH_CLERK	0
23	Dilly	SH_CLERK	0
24	Doran	SA_REP	D
25	Ernst	IT_PROG	C
26	Errazuriz	SA_MAN	0
27	Everett	SH_CLERK	0
28	Faviet	FI_ACCOUNT	0
29	Fay	MK_REP	0
30	Feeney	SH_CLERK	0
31	Fleaur	SH_CLERK	0
32	Fox	SA_REP	D
33	Fripp	ST_MAN	B
34	Gates	SH_CLERK	0
35	Gee	ST_CLERK	E
36	Geoni	SH_CLERK	0
37	Gietz	AC_ACCOUNT	0
38	Grant	SH_CLERK	0
39	Grant	SA_REP	D
40	Greenberg	FI_MGR	0
41	Greene	SA_REP	D
42	Hall	SA_REP	D
43	Hartstein	MK_MAN	0
44	Higgins	AC_MGR	0
45	Himuro	PU_CLERK	0
46	Hunold	IT_PROG	C
47	Hutton	SA_REP	D
48	Johnson	SA_REP	D
49	Jones	SH_CLERK	0
50	Kaufling	ST_MAN	B
51	Khoo	PU_CLERK	0
52	King	SA_REP	D
53	King	AD_PRES	A
54	Kochhar	AD_VP	0
55	Kumar	SA_REP	D
56	Ladwig	ST_CLERK	E
57	Landry	ST_CLERK	E
58	Lee	SA_REP	D
59	Livingston	SA_REP	D
60	Lorentz	IT_PROG	C
61	Mallin	ST_CLERK	E
62	Markle	ST_CLERK	E
63	Marlow	ST_CLERK	E
64	Marvins	SA_REP	D
65	Matos	ST_CLERK	E
66	Mavris	HR_REP	0
67	McCain	SH_CLERK	0
68	McEwen	SA_REP	D
69	Mikkili...	ST_CLERK	E
70	Mourgos	ST_MAN	B
71	Nayer	ST_CLERK	E
72	OConnell	SH_CLERK	0
73	Olsen	SA_REP	D
74	Olson	ST_CLERK	E
75	Ozer	SA_REP	D
76	Partners	SA_MAN	0
77	Pataballa	IT_PROG	C
78	Patel	ST_CLERK	E
79	Perkins	SH_CLERK	0
80	Philtanker	ST_CLERK	E
81	Popp	FI_ACCOUNT	0
82	Rajs	ST_CLERK	E
83	Raphaely	PU_MAN	0
84	Rogers	ST_CLERK	E
85	Russell	SA_MAN	0
86	Sarchand	SH_CLERK	0
87	Sciarra	FI_ACCOUNT	0
88	Seo	ST_CLERK	E
89	Sewall	SA_REP	D
90	Smith	SA_REP	D
91	Smith	SA_REP	D
92	Stiles	ST_CLERK	E
93	Sullivan	SH_CLERK	0
94	Sully	SA_REP	D
95	Taylor	SA_REP	D
96	Taylor	SH_CLERK	0
97	Tobias	PU_CLERK	0
98	Tucker	SA_REP	D
99	Tuvault	SA_REP	D
100	Urman	FI_ACCOUNT	0
101	Vargas	ST_CLERK	E
102	Vishney	SA_REP	D
103	Vollman	ST_MAN	B
104	Walsh	SH_CLERK	0
105	Weiss	ST_MAN	B
106	Whalen	AD_ASST	0
107	Zlotkey	SA_MAN	0

QUEZON CITY UNIVERSITY

COLLEGE OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

14. Display the last name, hire date, and day of the week on which the employee started. Label the column DAY. Order the results by the day of the week, starting with Monday.

```
SELECT last_name,
       hire_date,
       TO_CHAR(hire_date, 'Day') AS DAY
FROM employees
ORDER BY TO_CHAR(hire_date, 'D');
```

Query Result | SQL | Fetched 50 rows in 0.005 seconds

	LAST_NAME	HIRE_DATE	DAY
1	Pataballa	05-FEB-06	Sunday
2	Landry	14-JAN-07	Sunday
3	Grant	13-JAN-08	Sunday
4	Tucker	30-JAN-05	Sunday
5	McEwen	01-AUG-04	Sunday
6	Fripp	10-APR-05	Sunday
7	Weiss	18-JUL-04	Sunday
8	Tobias	24-JUL-05	Sunday
9	Vargas	09-JUL-06	Sunday
10	Livingston	23-APR-06	Sunday
11	Bull	20-FEB-05	Sunday
12	Khoo	18-MAY-03	Sunday
13	Geoni	03-FEB-08	Sunday
14	Seo	12-FEB-06	Sunday
15	Atkinson	30-OCT-05	Sunday
16	Ernst	21-MAY-07	Monday
17	Walsh	24-APR-06	Monday
18	Vollman	10-OCT-05	Monday
19	Kumar	21-APR-08	Monday
20	Banda	21-APR-08	Monday
21	Ande	24-MAR-08	Monday
22	Greene	19-MAR-07	Monday
23	Cambrault	15-OCT-07	Monday
24	Ladwig	14-JUL-03	Monday

	LAST_NAME	HIRE_DATE	DAY
25	Mallin	14-JUN-04	Monday
26	Chung	14-JUN-05	Tuesday
27	Gates	11-JUL-06	Tuesday
28	Fox	24-JAN-06	Tuesday
29	Zlotkey	29-JAN-08	Tuesday
30	Feeney	23-MAY-06	Tuesday
31	Sarchand	27-JAN-04	Tuesday
32	Olson	10-APR-07	Tuesday
33	Urman	07-MAR-06	Tuesday
34	Abel	11-MAY-04	Tuesday
35	Hunold	03-JAN-06	Tuesday
36	King	17-JUN-03	Tuesday
37	Hartstein	17-FEB-04	Tuesday
38	Taylor	24-JAN-06	Tuesday
39	Whalen	17-SEP-03	Wednesday
40	Bell	04-FEB-04	Wednesday
41	Perkins	19-DEC-07	Wednesday
42	Cabrio	07-FEB-07	Wednesday
43	Partners	05-JAN-05	Wednesday
44	Matos	15-MAR-06	Wednesday
45	Stiles	26-OCT-05	Wednesday
46	Philtanker	06-FEB-08	Wednesday
47	Gee	12-DEC-07	Wednesday
48	Marlow	16-FEB-05	Wednesday
49	Himuro	15-NOV-06	Wednesday
50	Chen	28-SEP-05	Wednesday
51	Lorentz	07-FEB-07	Wednesday
52	Fay	17-AUG-05	Wednesday

52	Fay	17-AUG-05	Wednesday
53	Kochhar	21-SEP-05	Wednesday
54	Kaufling	01-MAY-03	Thursday
55	Mikkili...	28-SEP-06	Thursday
56	Patel	06-APR-06	Thursday
57	Errazuriz	10-MAR-05	Thursday
58	Bernstein	24-MAR-05	Thursday
59	Olsen	30-MAR-06	Thursday
60	Sully	04-MAR-04	Thursday
61	Smith	10-MAR-05	Thursday
62	Doran	15-DEC-05	Thursday
63	Marvins	24-JAN-08	Thursday
64	Bloom	23-MAR-06	Thursday
65	Grant	24-MAY-07	Thursday
66	Fleaur	23-FEB-06	Thursday
67	Sullivan	21-JUN-07	Thursday
68	Everett	03-MAR-05	Thursday
69	OConnell	21-JUN-07	Thursday
70	Higgins	07-JUN-02	Friday
71	Sciarra	30-SEP-05	Friday
72	Mavris	07-JUN-02	Friday
73	Mourgos	16-NOV-07	Friday
74	Russell	01-OCT-04	Friday
75	Colmenares	10-AUG-07	Friday
76	Rajs	17-OCT-03	Friday
77	Johnson	04-JAN-08	Friday
78	Taylor	24-MAR-06	Friday
79	Faviet	16-AUG-02	Friday

80	Gietz	07-JUN-02	Friday
81	Smith	23-FEB-07	Friday
82	Ozer	11-MAR-05	Friday
83	Popp	07-DEC-07	Friday
84	Vishney	11-NOV-05	Friday
85	Sewall	03-NOV-06	Friday
86	King	30-JAN-04	Friday
87	Tuvault	23-NOV-07	Friday
88	Baer	07-JUN-02	Friday
89	Jones	17-MAR-07	Saturday
90	McCain	01-JUL-06	Saturday
91	Dilly	13-AUG-05	Saturday
92	Dellinger	24-JUN-06	Saturday
93	Hutton	19-MAR-05	Saturday
94	Bates	24-MAR-07	Saturday
95	Lee	23-FEB-08	Saturday
96	Cambrault	09-DEC-06	Saturday
97	Hall	20-AUG-05	Saturday
98	Davies	29-JAN-05	Saturday
99	Rogers	26-AUG-06	Saturday
100	Bissot	20-AUG-05	Saturday
101	Markle	08-MAR-08	Saturday
102	Nayer	16-JUL-05	Saturday
103	Baida	24-DEC-05	Saturday
104	Raphaely	07-DEC-02	Saturday
105	Greenberg	17-AUG-02	Saturday
106	De Haan	13-JAN-01	Saturday
107	Austin	25-JUN-05	Saturday

15. The HR department needs a report to display the employee number, last name, salary, and salary increased by 15.5% (expressed as a whole number) for each employee. Label the column New Salary.

```
SELECT employee_id,
       last_name,
       salary,
       ROUND(salary * 1.155) AS "New Salary"
FROM employees;
```

Query Result | SQL | Fetched 50 rows in 0.005 seconds

	EMPLOYEE_ID	LAST_NAME	SALARY	New Salary
1	100	King	24000	27720
2	101	Kochhar	17000	19635
3	102	De Haan	17000	19635
4	103	Hunold	9000	10395
5	104	Ernst	6000	6930
6	105	Austin	4800	5544
7	106	Pataballa	4800	5544
8	107	Lorentz	4200	4851
9	108	Greenberg	12008	13869
10	109	Faviet	9000	10395
11	110	Chen	8200	9471
12	111	Sciarra	7700	8894
13	112	Urman	7800	9009
14	113	Popp	6900	7970
15	114	Raphaely	11000	12705
16	115	Khoo	3100	3581
17	116	Baida	2900	3350
18	117	Tobias	2800	3234
19	118	Himuro	2600	3003
20	119	Colmenares	2500	2888
21	120	Weiss	8000	9240
22	121	Fripp	8200	9471
23	122	Kaufling	7900	9125
24	123	Vollman	6500	7508

25	124	Mourgos	5800	6699
26	125	Nayer	3200	3696
27	126	Mikkilineni	2700	3119
28	127	Landry	2400	2772
29	128	Markle	2200	2541
30	129	Bissot	3300	3812
31	130	Atkinson	2800	3234
32	131	Marlow	2500	2888
33	132	Olson	2100	2426
34	133	Mallin	3300	3812
35	134	Rogers	2900	3350
36	135	Gee	2400	2772
37	136	Philtanker	2200	2541
38	137	Ladwig	3600	4158
39	138	Stiles	3200	3696
40	139	Seo	2700	3119
41	140	Patel	2500	2888
42	141	Rajs	3500	4043
43	142	Davies	3100	3581
44	143	Matos	2600	3003
45	144	Vargas	2500	2888
46	145	Russell	14000	16170
47	146	Partners	13500	15593
48	147	Errazuriz	12000	13860
49	148	Cambrault	11000	12705

50	149	Zlotkey	10500	12128
51	150	Tucker	10000	11550
52	151	Bernstein	9500	10973
53	152	Hall	9000	10395
54	153	Olsen	8000	9240
55	154	Cambrault	7500	8663
56	155	Tuvault	7000	8085
57	156	King	10000	11550
58	157	Sully	9500	10973
59	158	McEwen	9000	10395
60	159	Smith	8000	9240
61	160	Doran	7500	8663
62	161	Sewall	7000	8085
63	162	Vishney	10500	12128
64	163	Greene	9500	10973
65	164	Marvins	7200	8316
66	165	Lee	6800	7854
67	166	Ande	6400	7392
68	167	Banda	6200	7161
69	168	Ozer	11500	13283
70	169	Bloom	10000	11550
71	170	Fox	9600	11088
72	171	Smith	7400	8547
73	172	Bates	7300	8432
74	173	Kumar	6100	7046
75	174	Abel	11000	12705
76	175	Hutton	8800	10164

76	175	Hutton	8800	10164
77	176	Taylor	8600	9933
78	177	Livingston	8400	9702
79	178	Grant	7000	8085
80	179	Johnson	6200	7161
81	180	Taylor	3200	3696
82	181	Fleaur	3100	3581
83	182	Sullivan	2500	2888
84	183	Geoni	2800	3234
85	184	Sarchand	4200	4851
86	185	Bull	4100	4736
87	186	Dellinger	3400	3927
88	187	Cabrio	3000	3465
89	188	Chung	3800	4389
90	189	Dilly	3600	4158
91	190	Gates	2900	3350
92	191	Perkins	2500	2888
93	192	Bell	4000	4620
94	193	Everett	3900	4505
95	194	McCain	3200	3696
96	195	Jones	2800	3234
97	196	Walsh	3100	3581
98	197	Feeney	3000	3465
99	198	OConnell	2600	3003
100	199	Grant	2600	3003
101	200	Whalen	4400	5082
102	201	Hartstein	13000	15015
103	202	Fay	6000	6930

103	202	Fay	6000	6930
104	203	Mavris	6500	7508
105	204	Baer	10000	11550
106	205	Higgins	12008	13869
107	206	Gietz	8300	9587

16. Create a query to display the last name and the number of weeks employed for all employees in department 90. Label the number of weeks column as TENURE. Truncate the number of weeks value to 0 decimal places. Show the records in descending order of the employees' tenure.

```
SELECT last_name,
       TRUNC(MONTHS_BETWEEN(SYSDATE, hire_date) * 4.33, 0) AS TENURE
FROM employees
WHERE department_id = 90
ORDER BY TENURE DESC;
```

Query Result | SQL | All Rows Fetched: 3 in 0.003 seconds

	LAST_NAME	TENURE
1	De Haan	1251
2	King	1125
3	Kochhar	1008

17. For every department, find the differences between the highest and lowest salaries. Label the column DIFFERENCE. Order by department id in ascending order.

```
SELECT department_id,
       MAX(salary) - MIN(salary) AS DIFFERENCE
FROM employees
GROUP BY department_id
ORDER BY department_id ASC;
```

Query Result | SQL | All Rows Fetched: 12 in 0.031 seconds

	DEPARTMENT_ID	DIFFERENCE
1	10	0
2	20	7000
3	30	8500
4	40	0
5	50	6100
6	60	4800
7	70	0
8	80	7900
9	90	7000
10	100	5108
11	110	3708
12	(null)	0

18. The HR department needs a report of employees in Toronto. Display the last name, job title, department number, and the department name for all employees who work in Toronto. Use WHERE clause to join the tables.

```
SELECT e.last_name,
       j.job_title,
       e.department_id,
       d.department_name
FROM employees e, jobs j, departments d, locations l
WHERE e.job_id = j.job_id
AND e.department_id = d.department_id
AND d.location_id = l.location_id
AND l.city = 'Toronto';
```

Query Result | SQL | All Rows Fetched: 2 in 0.007 seconds

	LAST_NAME	JOB_TITLE	DEPARTMENT_ID	DEPARTMENT_NAME
1	Hartstein	Marketing Manager	20	Marketing
2	Fay	Marketing Representative	20	Marketing

QUEZON CITY UNIVERSITY

COLLEGE OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

19. Display the employee id, last name, job title, department name, and years of service of all employees who worked in the company for more than 3 years. Join the tables using WHERE or JOIN.ON.

```
SELECT e.employee_id,
       e.last_name,
       j.job_title,
       d.department_name,
       FLOOR(MONTHS_BETWEEN(SYSDATE, e.hire_date) / 12) AS Years_of_Service
FROM employees e
JOIN jobs j ON e.job_id = j.job_id
JOIN departments d ON e.department_id = d.department_id
WHERE MONTHS_BETWEEN(SYSDATE, e.hire_date) / 12 > 3;
```

Query Result | SQL | Fetched 50 rows in 0.011 seconds

	EMPLOYEE_ID	LAST_NAME	JOB_TITLE	DEPARTMENT_NAME	YEARS_OF_SERVICE
1	200	Whalen	Administration Assistant	Administration	21
2	202	Fay	Marketing Representative	Marketing	19
3	201	Hartstein	Marketing Manager	Marketing	20
4	114	Raphaely	Purchasing Manager	Purchasing	22
5	119	Colmenares	Purchasing Clerk	Purchasing	17
6	118	Himuro	Purchasing Clerk	Purchasing	18
7	117	Tobias	Purchasing Clerk	Purchasing	19
8	115	Khoo	Purchasing Clerk	Purchasing	21
9	116	Baida	Purchasing Clerk	Purchasing	19
10	203	Mavris	Human Resources Representative	Human Resources	22
11	120	Weiss	Stock Manager	Shipping	20
12	121	Fripp	Stock Manager	Shipping	19
13	122	Kaufling	Stock Manager	Shipping	21
14	123	Vollman	Stock Manager	Shipping	19
15	124	Mourgos	Stock Manager	Shipping	17
16	139	Seo	Stock Clerk	Shipping	19
17	138	Stiles	Stock Clerk	Shipping	19
18	137	Ladwig	Stock Clerk	Shipping	21
19	136	Philtanker	Stock Clerk	Shipping	17
20	135	Gee	Stock Clerk	Shipping	17
21	134	Rogers	Stock Clerk	Shipping	18
47	192	Bell	Shipping Clerk	Shipping	21
48	191	Perkins	Shipping Clerk	Shipping	17
49	190	Gates	Shipping Clerk	Shipping	18
50	189	Dilly	Shipping Clerk	Shipping	19
51	188	Chung	Shipping Clerk	Shipping	19
52	187	Cabrio	Shipping Clerk	Shipping	18
53	186	Dellinger	Shipping Clerk	Shipping	18
54	185	Bull	Shipping Clerk	Shipping	19
55	184	Sarchand	Shipping Clerk	Shipping	21
56	105	Austin	Programmer	IT	19
57	106	Pataballa	Programmer	IT	19
58	107	Lorentz	Programmer	IT	18
59	104	Ernst	Programmer	IT	17
60	103	Hunold	Programmer	IT	19
61	204	Baer	Public Relations Representa...	Public Relations	22
62	162	Vishney	Sales Representative	Sales	19
63	150	Tucker	Sales Representative	Sales	20
64	151	Bernstein	Sales Representative	Sales	19
65	152	Hall	Sales Representative	Sales	19
66	153	Olsen	Sales Representative	Sales	18
67	154	Cambrault	Sales Representative	Sales	18
68	177	Livingston	Sales Representative	Sales	18
69	176	Taylor	Sales Representative	Sales	18
70	175	Hutton	Sales Representative	Sales	19
71	174	Abel	Sales Representative	Sales	20
72	173	Kumar	Sales Representative	Sales	16
73	172	Bates	Sales Representative	Sales	17
22	133	Mallin	Stock Clerk	Shipping	20
23	144	Vargas	Stock Clerk	Shipping	18
24	131	Marlow	Stock Clerk	Shipping	19
25	130	Atkinson	Stock Clerk	Shipping	19
26	129	Bissot	Stock Clerk	Shipping	19
27	128	Markle	Stock Clerk	Shipping	16
28	127	Landry	Stock Clerk	Shipping	18
29	126	Mikkilineni	Stock Clerk	Shipping	18
30	125	Nayer	Stock Clerk	Shipping	19
31	140	Patel	Stock Clerk	Shipping	18
32	141	Rajs	Stock Clerk	Shipping	21
33	142	Davies	Stock Clerk	Shipping	20
34	143	Matos	Stock Clerk	Shipping	18
35	132	Olson	Stock Clerk	Shipping	17
36	183	Geoni	Shipping Clerk	Shipping	17
37	182	Sullivan	Shipping Clerk	Shipping	17
38	181	Fleaur	Shipping Clerk	Shipping	18
39	180	Taylor	Shipping Clerk	Shipping	19
40	199	Grant	Shipping Clerk	Shipping	17
41	198	OConnell	Shipping Clerk	Shipping	17
42	197	Feeney	Shipping Clerk	Shipping	18
43	196	Walsh	Shipping Clerk	Shipping	18
44	195	Jones	Shipping Clerk	Shipping	17
45	194	McCain	Shipping Clerk	Shipping	18
46	193	Everett	Shipping Clerk	Shipping	19
74	171	Smith	Sales Representative	Sales	17
75	170	Fox	Sales Representative	Sales	19
76	169	Bloom	Sales Representative	Sales	18
77	168	Ozer	Sales Representative	Sales	19
78	167	Banda	Sales Representative	Sales	16
79	166	Ande	Sales Representative	Sales	16
80	165	Lee	Sales Representative	Sales	16
81	164	Marvins	Sales Representative	Sales	17
82	163	Greene	Sales Representative	Sales	17
83	179	Johnson	Sales Representative	Sales	17
84	161	Sewall	Sales Representative	Sales	18
85	160	Doran	Sales Representative	Sales	19
86	159	Smith	Sales Representative	Sales	19
87	158	McEwen	Sales Representative	Sales	20
88	157	Sully	Sales Representative	Sales	20
89	156	King	Sales Representative	Sales	21
90	155	Tuvault	Sales Representative	Sales	17
91	149	Zlotkey	Sales Manager	Sales	17
92	145	Russell	Sales Manager	Sales	20
93	146	Partners	Sales Manager	Sales	20
94	147	Errazuriz	Sales Manager	Sales	19
95	148	Cambrault	Sales Manager	Sales	17
96	101	Kochhar	Administration Vice President	Executive	19
97	102	De Haan	Administration Vice President	Executive	24
98	100	King	President	Executive	21
99	108	Greenberg	Finance Manager	Finance	22
100	112	Urman	Accountant	Finance	18
101	111	Sciarra	Accountant	Finance	19
102	113	Popp	Accountant	Finance	17
103	109	Faviet	Accountant	Finance	22
104	110	Chen	Accountant	Finance	19
105	205	Higgins	Accounting Manager	Accounting	22
106	206	Gietz	Public Accountant	Accounting	22

20. Display the employee id, last name, job title, city, country name of employees not from Europe or Americas. Use JOIN..ON to join the tables.

```
SELECT
    e.employee_id,
    e.last_name,
    j.job_title,
    l.city,
    c.country_name
FROM HR.employees e
JOIN HR.jobs j ON e.job_id = j.job_id
JOIN HR.departments d ON e.department_id = d.department_id
JOIN HR.locations l ON d.location_id = l.location_id
JOIN HR.countries c ON l.country_id = c.country_id
JOIN HR.regions r ON c.region_id = r.region_id
WHERE r.region_name NOT IN ('Europe', 'Americas');
```

Query Result

SQL | All Rows Fetched: 0 in 0.004 seconds

EMPLOYE...	LAST_NAME	JOB_TITLE	CITY	COUNTRY...