

SQL Queries

Based on Employee and Department tables, perform the following SQL queries:

1. Write a query in SQL to display the last name and job title of all employees who do not have a manager

SQL Query

```
SELECT last_name, job_title
FROM employees, jobs
WHERE manager_id IS NULL and employees.job_id = jobs.job_id;
```

OUTPUT

LAST_NAME	JOB_TITLE
King	President
1 rows returned in 0.01 seconds Download	

2. Write a query in SQL to display the last name, salary, and commission of all employees who earn commissions. Sort data in descending order of salary and commissions.

SQL Query

```
SELECT last_name, salary, commission_pct
FROM employees
WHERE commission_pct IS NOT NULL
ORDER BY salary DESC, commission_pct DESC;
```

OUTPUT

LAST_NAME	SALARY	COMMISSION_PCT
Russell	14000	.4
Partners	13500	.3
Errazuriz	12000	.3
Ozer	11500	.25
Cambrault	11000	.3
Abel	11000	.3
Vishney	10500	.25
Zlotkey	10500	.2
King	10000	.35
Tucker	10000	.3
Bloom	10000	.2
Fox	9600	.2
Sully	9500	.35
Bernstein	9500	.25
Greene	9500	.15
McEwen	9000	.35

3. Write a query in SQL that prompts the user for a manager ID and generates the employee ID, last name, salary, and department for that manager's employees. The HR department wants the ability to sort the report on a selected column.

SQL Query

```
SELECT e.manager_id, e.employee_id, e.last_name, salary, e.department_id, d.department_name
FROM employees e, departments d
WHERE e.manager_id = &manager_num and e.department_id = d.department_id
ORDER BY '&column';
```

INPUT

manager_num = 100 and column = 'last_name'

OUTPUT

MANAGER_ID	EMPLOYEE_ID	LAST_NAME	SALARY	DEPARTMENT_ID	DEPARTMENT_NAME
100	148	Cambrault	11000	80	Sales
100	102	De Haan	17000	90	Executive
100	147	Errazuriz	12000	80	Sales
100	121	Fripp	8200	50	Shipping
100	201	Hartstein	13000	20	Marketing
100	122	Kaufling	7900	50	Shipping
100	101	Kochhar	17000	90	Executive
100	124	Mourgos	5800	50	Shipping
100	146	Partners	13500	80	Sales
100	114	Raphaely	11000	30	Purchasing
100	145	Russell	14000	80	Sales
100	123	Vollman	6500	50	Shipping
100	120	Weiss	8000	50	Shipping
100	149	Zlotkey	10500	80	Sales

4. Write a query in SQL to Display all employee last names in which the third letter of the name is a.

SQL Query

```
SELECT last_name
FROM employees
WHERE last_name LIKE '__a%';
```

OUTPUT

LAST_NAME
Grant
Grant
Whalen

3 rows returned in 0.01 seconds [Download](#)

5. Write a query in SQL to Display the last name of all employees who have both an a and an e in their last name.

SQL Query

```
SELECT last_name
FROM employees
WHERE last_name LIKE '%a%' and last_name LIKE '%e%';
```

OUTPUT

Results	Explain	Describe	Saved SQL	History
LAST_NAME				
De Haan				
Faviet				
Raphaely				
Colmenares				
Nayer				
Markle				
Philtanker				
Patel				
Davies				
Partners				
Sewall				
Bates				
Fleur				
Gates				
Whalen				
Hartstein				

6. Write a query in SQL to Display the last name, job, and salary for all employees whose job is sales representative or stock clerk and whose salary is not equal to \$2,500, \$3,500, or \$7,000.

SQL Query

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

OUTPUT

LAST_NAME	JOB_ID	SALARY
Nayer	ST_CLERK	3200
Mikkilineni	ST_CLERK	2700
Landry	ST_CLERK	2400
Markle	ST_CLERK	2200
Bissot	ST_CLERK	3300
Atkinson	ST_CLERK	2800
Olson	ST_CLERK	2100
Mallin	ST_CLERK	3300
Rogers	ST_CLERK	2900
Gee	ST_CLERK	2400
Philtanker	ST_CLERK	2200
Ladwig	ST_CLERK	3600
Stiles	ST_CLERK	3200
Seo	ST_CLERK	2700
Davies	ST_CLERK	3100
Matos	ST_CLERK	2600

7. Write a query in SQL 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.

SQL Query

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

OUTPUT

EMPLOYEE_ID	LAST_NAME	SALARY	New Salary
100	King	24000	27720
101	Kochhar	17000	19635
102	De Haan	17000	19635
103	Hunold	9000	10395
104	Ernst	6000	6930
105	Austin	4800	5544
106	Pataballa	4800	5544
107	Lorentz	4200	4851
108	Greenberg	12008	13869
109	Faviet	9000	10395
110	Chen	8200	9471
111	Sciarra	7700	8894
112	Urman	7800	9009
113	Popp	6900	7970
114	Raphaely	11000	12705
115	Khoo	3100	3581

8. Create a report that produces the following for each employee: <employee last name> earns <salary> monthly but wants <3 times salary>. Label the column Dream Salaries.

SQL Query

```
SELECT last_name||' earns '||salary||' monthly but wants '||3*salary AS "Dream Salaries"  
FROM employees;
```

OUTPUT

Dream Salaries
King earns 24000 monthly but wants 72000
Kochhar earns 17000 monthly but wants 51000
De Haan earns 17000 monthly but wants 51000
Hunold earns 9000 monthly but wants 27000
Ernst earns 6000 monthly but wants 18000
Austin earns 4800 monthly but wants 14400
Pataballa earns 4800 monthly but wants 14400
Lorentz earns 4200 monthly but wants 12600
Greenberg earns 12008 monthly but wants 36024
Faviet earns 9000 monthly but wants 27000
Chen earns 8200 monthly but wants 24600
Sciarra earns 7700 monthly but wants 23100
Urman earns 7800 monthly but wants 23400
Popp earns 6900 monthly but wants 20700
Raphaely earns 11000 monthly but wants 33000
Khoo earns 3100 monthly but wants 9300

9. Create a query to display the last name and salary for all employees. Format the salary to be 15 characters long, left-padded with the \$ symbol. Label the column SALARY.

SQL Query

```
SELECT last_name,LPAD(salary,15,'$') AS "SALARY"  
FROM employees;
```

OUTPUT

LAST_NAME	SALARY
King	\$\$\$\$\$\$\$\$\$24000
Kochhar	\$\$\$\$\$\$\$\$\$17000
De Haan	\$\$\$\$\$\$\$\$\$17000
Hunold	\$\$\$\$\$\$\$\$\$9000
Ernst	\$\$\$\$\$\$\$\$\$6000
Austin	\$\$\$\$\$\$\$\$\$4800
Pataballa	\$\$\$\$\$\$\$\$\$4800
Lorentz	\$\$\$\$\$\$\$\$\$4200
Greenberg	\$\$\$\$\$\$\$\$\$12008
Faviet	\$\$\$\$\$\$\$\$\$9000
Chen	\$\$\$\$\$\$\$\$\$8200
Sciarra	\$\$\$\$\$\$\$\$\$7700
Urman	\$\$\$\$\$\$\$\$\$7800
Popp	\$\$\$\$\$\$\$\$\$6900
Raphaely	\$\$\$\$\$\$\$\$\$11000
Khoo	\$\$\$\$\$\$\$\$\$3100

10. Display each employee's last name, hire date, and salary review date, which is the first Monday after six months of service. Label the column REVIEW. Format the dates to appear in the format similar to "Monday, the Thirty-First of July, 2000."

SQL Query

```
SELECT last_name,hire_date, TO_CHAR((NEXT_DAY(ADD_MONTHS(hire_date,6),'MONDAY')),'Day, "the" Ddspth "of" Month, YYYY') AS "REVIEW"
FROM employees;
```

OUTPUT

LAST_NAME	HIRE_DATE	REVIEW
King	06/17/2003	Monday, the Twenty-Second of December, 2003
Kochhar	09/21/2005	Monday, the Twenty-Seventh of March, 2006
De Haan	01/13/2001	Monday, the Sixteenth of July, 2001
Hunold	01/03/2006	Monday, the Tenth of July, 2006
Ernst	05/21/2007	Monday, the Twenty-Sixth of November, 2007
Austin	06/25/2005	Monday, the Twenty-Sixth of December, 2005
Pataballa	02/05/2006	Monday, the Seventh of August, 2006
Lorentz	02/07/2007	Monday, the Thirteenth of August, 2007
Greenberg	08/17/2002	Monday, the Twenty-Fourth of February, 2003
Faviet	08/16/2002	Monday, the Seventeenth of February, 2003
Chen	09/28/2005	Monday, the Third of April, 2006
Sciarra	09/30/2005	Monday, the Third of April, 2006
Urman	03/07/2006	Monday, the Eleventh of September, 2006
Popp	12/07/2007	Monday, the Ninth of June, 2008
Raphaely	12/07/2002	Monday, the Ninth of June, 2003
Khoo	05/18/2003	Monday, the Twenty-Fourth of November, 2003

11. 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.

SQL Query

```
SELECT last_name,hire_date,TO_CHAR(hire_date,'Day') AS "DAY"
FROM employees
ORDER BY TO_CHAR(hire_date-1,'d');
```

OUTPUT

LAST_NAME	HIRE_DATE	DAY
Ladwig	07/14/2003	Monday
Cambrault	10/15/2007	Monday
Mallin	06/14/2004	Monday
Ernst	05/21/2007	Monday
Greene	03/19/2007	Monday
Banda	04/21/2008	Monday
Walsh	04/24/2006	Monday
Ande	03/24/2008	Monday
Vollman	10/10/2005	Monday
Kumar	04/21/2008	Monday
Olson	04/10/2007	Tuesday
Urman	03/07/2006	Tuesday
Hunold	01/03/2006	Tuesday
King	06/17/2003	Tuesday
Hartstein	02/17/2004	Tuesday
Feeney	05/23/2006	Tuesday