

Extra SQL Exercises

EMP:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	-	11/17/2001	82000	-	10
7782	CLARK	MANAGER	7839	06/09/2009	27500	-	10
7934	MILLER	CLERK	7782	01/23/2006	13200	-	10
7566	JONES	MANAGER	7839	04/02/2012	26850	-	20
7788	SCOTT	ANALYST	7566	12/09/2008	19500	-	20
7902	FORD	ANALYST	7566	12/03/2014	21500	-	20
7369	SMITH	CLERK	7902	12/17/2002	13750	-	20
7876	ADAMS	CLERK	7788	01/12/2011	11900	-	20
7698	BLAKE	MANAGER	7839	05/01/2005	24000	-	30
7499	ALLEN	SALESMAN	7698	02/20/2006	19000	6300	30
7521	WARD	SALESMAN	7698	02/22/2018	18500	4500	30
7654	MARTIN	SALESMAN	7698	09/28/2008	15650	3400	30
7844	TURNER	SALESMAN	7698	09/08/2011	18500	0	30
7900	JAMES	CLERK	7698	12/03/1998	11950	-	30
4422	STEVENS	MANAGER	7839	02/14/2019	24750	-	50
3258	GREEN	SALESMAN	4422	07/24/2003	18200	2750	50
6548	BARNES	CLERK	4422	01/16/2018	11950	-	50

DEPT:

DEPTNO	DNAME	LOC
10	ACCOUNTING	LONDON
20	RESEARCH	PRESTON
30	SALES	LIVERPOOL
40	OPERATIONS	MANCHESTER
50	MARKETING	LUTON

Subqueries

SQL allows us to write SELECT statements that contain other SELECT statements. These are known as a **subquery** or a **nested query**.

Let's say we want to find the name and job of the employee with the lowest salary. This seems an easy enough request, and the following statement seems correct. However, it will give us an error:

```
SELECT ename, job, MIN(sal) FROM emp
```

We have seen this before; we are mixing a group function (MIN) with non-group columns (ename, job), which is not allowed in SQL.

To solve this query, we firstly need to find the minimum salary, and then to go through the table and check each row to find the name of the employee who has the salary equal to the minimum salary:

```
SELECT MIN(sal) FROM emp
```

```
SELECT ename, job, sal FROM emp
```

Clearly, we want to be able to combine the two queries into one query. We can do this with a subquery:

```
SELECT ename, job, sal FROM emp  
WHERE sal = (SELECT MIN(sal) FROM emp)
```

Now we can see that the employee with the lowest salary is a Clerk named Adams who 's salary is £11900. The inner (subquery) is executed first which returns the value 11900. Then the outer query executes and selects the employee with a salary that matches 11900.

Now let's say we want to find all employees that earn more than TURNER:

```
SELECT ename, sal FROM emp  
WHERE sal > (SELECT sal FROM emp WHERE ename = 'TURNER')
```

Firstly, the subquery executes and finds Turner's salary (£18,500). Then the outer query selects all rows that have a salary greater than £18,500.

Now have a go at these:

Date Functions

SQL has a variety of date functions:

Function	Example	Result	Description
ADD_MONTHS	ADD_MONTHS (DATE '2016-02-29', 1)	31-MAR-16	Add a number of months (n) to a date and return the same day which is n of months away.
CURRENT_DATE	SELECT CURRENT_DATE FROM dual	06-AUG-2017	Return the current

Function	Example	Result	Description
		19:43:44	date and time in the session time zone
CURRENT_TIMESTAMP	SELECT CURRENT_TIMESTAMP FROM dual	06-AUG-17 08.26.52.742000000 PM -07:00	Return the current date and time with time zone in the session time zone
DBTIMEZONE	SELECT DBTIMEZONE FROM dual;	-07:00	Get the current database time zone
EXTRACT	EXTRACT(YEAR FROM SYSDATE)	2017	Extract a value of a date time field e.g., YEAR, MONTH, DAY, ... from a date time value.
FROM_TZ	FROM_TZ(TIMESTAMP '2017-08-08 08:09:10', '-09:00')	08-AUG-17 08.09.10.000000000 AM -07:00	Convert a timestamp and a time zone to a TIMESTAMP WITH TIME ZONE value
LAST_DAY	LAST_DAY(DATE '2016-02-01')	29-FEB-16	Gets the last day of the month of a specified date.
LOCALTIMESTAMP	SELECT LOCALTIMESTAMP FROM dual	06-AUG-17 08.26.52.742000000 PM	Return a TIMESTAMP value that represents the current date and time in the session time zone.
MONTHS_BETWEEN	MONTHS_BETWEEN(DATE '2017-07-01', DATE '2017-01-01')	6	Return the number of months between two dates.
NEW_TIME	NEW_TIME(TO_DATE('08-07-2017 01:30:45', 'MM-DD-YYYY HH24:MI:SS'), 'AST', 'PST')	06-AUG-2017 21:30:45	Convert a date in one time zone to another

Function	Example	Result	Description
NEXT_DAY	<code>NEXT_DAY (DATE '2000-01-01', 'SUNDAY')</code>	02-JAN-00	Get the first weekday that is later than a specified date.
ROUND	<code>ROUND (DATE '2017-07-16', 'MM')</code>	01-AUG-17	Return a date rounded to a specific unit of measure.
SESSIONTIMEZONE	<code>SELECT SESSIONTIMEZONE FROM dual;</code>	-07:00	Get the session time zone
SYSDATE	<code>SYSDATE</code>	01-AUG-17	Return the current system date and time of the operating system where the Oracle Database resides.
SYSTIMESTAMP	<code>SELECT SYSTIMESTAMP FROM dual;</code>	01-AUG-17 01.33.57.929000000 0 PM -07:00	Return the system date and time that includes fractional seconds and time zone.
TO_CHAR	<code>TO_CHAR (DATE '2017-01-01', 'DL')</code>	Sunday, January 01, 2017	Convert a DATE or an INTERVAL value to a character string in a specified format.
TO_DATE	<code>TO_DATE ('01 Jan 2017', 'DD MON YYYY')</code>	01-JAN-17	Convert a date which is in the character string to a <code>DATE</code> value
TRUNC	<code>TRUNC (DATE '2017-07-16', 'MM')</code>	01-JUL-17	Return a date truncated to a specific unit of measure.
TZ_OFFSET	<code>TZ_OFFSET ('Europe/London')</code>	+01:00	Get time zone offset of a time zone name

Function	Example	Result	Description
			from UTC

Exercises

1. Find the name and salary of the highest paid employee.
2. Find the names of all employees who work in the same department as WARD.
3. Find the names and salaries of all employees who earn more than BLAKE.
4. Find the department that has the most employees.
5. List the names and salaries of all employees who have a salary which is higher than the average salary of all employees.
6. List all employees who have a salary which is higher than the average salary for their department.
7. List the departments that do not have any analysts.
8. List the name and job of employees who are earning the highest salary for each job.
9. How many employees were hired in 2018.
10. How many employees have been hired within the last 20 years.