

LAB 2 SOLUTIONS

Exercise 1

Q1	List the first and last names of all employees.
	<pre>SELECT FIRSTNAME, LASTNAME FROM EMPLOYEES;</pre>
Q2	List all attributes of the projects with revenue greater than \$40,000.
	<pre>SELECT * FROM PROJECTS WHERE REVENUE>40000;</pre>
Q3	List the department codes of the projects with revenue between \$100,000 and \$150,000.
	<pre>SELECT DEPTCODE FROM PROJECTS WHERE REVENUE BETWEEN 100000 AND 150000;</pre>
Q4	List the project IDs for the projects that started on or before July 1, 2004.
	<pre>SELECT PROJECTID FROM PROJECTS WHERE STARTDATE <= '01/July/2004';</pre>
Q5	List the names of the departments that are top level (i.e., not a sub department).
	<pre>SELECT NAME FROM DEPARTMENTS WHERE SUBDEPTOF IS NULL;</pre>
Q6	List the ID and descriptions of the projects under the departments with code ACCNT, CNSLT, or HDWRE.
	<pre>SELECT PROJECTID, DESCRIPTION FROM PROJECTS WHERE DEPTCODE IN ('ACCNT ', 'CNSLT', 'HDWRE');</pre>
Q7	List all of the information about employees with last names that have exactly 8 characters and end in 'ware'.
	<pre>SELECT * FROM EMPLOYEES WHERE LASTNAME LIKE '____ware';</pre>
Q8	List the ID and last name of all employees who work for department ACTNG and make less than \$30,000.
	<pre>SELECT EMPLOYEEID, LASTNAME FROM EMPLOYEES WHERE DEPTCODE = 'ACTNG' AND SALARY >30000;</pre>
Q9	List the “magical” projects that have not started (indicated by a start date in the future or <i>NULL</i>) but are generating revenue.
	<pre>SELECT * FROM PROJECTS WHERE (STARTDATE IS NULL) OR (REVENUE >0 AND STARTDATE > GETDATE());</pre>
Q10	List the IDs of the projects either from the ACTNG department or that are ongoing end date). Exclude any projects that have revenue of \$50,000 or less.
	<pre>SELECT PROJECTID FROM PROJECTS WHERE (DEPTCODE = 'ACCNT' OR ENDDATE IS NULL) AND REVENUE>50000;</pre>

Exercise 2

Q1	List all employee names as one field called name.
	<code>SELECT FIRSTNAME+' '+LASTNAME AS "NAME" FROM EMPLOYEES;</code>
Q2	List all the department codes assigned to a project. Remove all duplicates.
	<code>SELECT DISTINCT DEPTCODE FROM PROJECTS;</code>
Q3	Find the project ID and duration of each project.
	<code>SELECT PROJECTID DATDIFF(DAY,STARTDATE, ENDDATE) AS "DURATION" FROM PROJECTS;</code>
Q4	Find the project ID and duration of each project. If the project has not finished, report its execution time as of now. [Hint: Getdate() gives current date]
	<code>Select projectid, FLOOR(CASE WHEN enddate is NULL THEN DATEDIFF(DAY,STARTDATE,GETDATE()) ELSE DATEDIFF(DAY,STARTDATE,ENDDATE) END) As duration from projects; OR SELECT PROJECTID, DATEDIFF(DAY,STARTDATE, COALESCE(ENDDATE,GETDATE())) AS DURATION FROM PROJECTS;</code>
Q5	For each completed project, find the project ID and average revenue per day.
	<code>Select projectid, revenue/datediff(day,STARTDATE,ENDDATE) as "average revenue" from projects where NOT enddate IS NULL;</code>
Q6	Find the years a project started. Remove duplicates.
	<code>Select distinct year(startdate) as "year" from projects;</code>

Q7	Find the IDs of employees assigned to a project that is more than 20 hours per week. Write three queries using 20, 40, and 60 hour work weeks.
	<p>Assumption -> Assigned Time = Number of working hours by a person on a project/ 40 Total working hours in a week is 40 In this query we first find the projectids on which employees worked more than given time (20 or 40 or 60) and then we output the employeeid of those projects. With 20 working hours:</p> <pre>SELECT DISTINCT EMPLOYEEID FROM WORKSON W1 WHERE PROJECTID IN (SELECT PROJECTID FROM WORKSON W2 GROUP BY PROJECTID HAVING SUM(ASSIGNEDTIME)*40 > 20) ;</pre> <p>With 40 working hours:</p> <pre>SELECT DISTINCT EMPLOYEEID FROM WORKSON W1 WHERE PROJECTID IN (SELECT PROJECTID FROM WORKSON W2 GROUP BY PROJECTID HAVING SUM(ASSIGNEDTIME)*40 > 40) ;</pre> <p>With 60 working hours</p> <pre>SELECT DISTINCT EMPLOYEEID FROM WORKSON W1 WHERE PROJECTID IN (SELECT PROJECTID FROM WORKSON W2 GROUP BY PROJECTID HAVING SUM(ASSIGNEDTIME)*40 > 60) ;</pre>
Q8	For each employee assigned to a task, output the employee ID with the following: <ul style="list-style-type: none"> • 'part time' if assigned time is < 0.33 • 'split time' if assigned time is >= 0.33 and < 0.67 • 'full time' if assigned time is >= 0.67
	<pre>SELECT EMPLOYEEID , (CASE WHEN ASSIGNEDTIME < 0.33 THEN 'part time' WHEN ASSIGNEDTIME > 0.33 AND ASSIGNEDTIME<0.67 THEN 'split time' WHEN ASSIGNEDTIME</pre>

	<pre>>= 0.67 THEN 'full time' END) AS "WORKTIME" FROM WORKSON;</pre>
--	---

Q9	We need to create a list of abbreviated project names. Each abbreviated name concatenates the first three characters of the project description, a hyphen, and the department code. All characters must be uppercase (e.g., EMP-ADMIN).
	<pre>Select UPPER(SUBSTRING(description,1,3)) + '-' + UPPER(deptcode) as "abbreviated" from projects;</pre>
Q10	For each project, list the ID and year the project started. Order the results in ascending order by year.
	<pre>Select projectid, year(startdate) as "year" from projects ORDER BY year;</pre>
Q11	If every employee is given a 5% raise, find the last name and new salary of the employees who will make more than \$50,000.
	<pre>SELECT LASTNAME, SALARY+(0.05*SALARY) AS "NEW SALARY" FROM EMPLOYEES WHERE SALARY+(0.05*SALARY) >50000;</pre>
Q12	For all the employees in the HDWRE department, list their ID, first name, last name, and salary after a 10% raise. The salary column in the result should be named Next Year.
	<pre>SELECT EMPLOYEEID, FIRSTNAME, LASTNAME, SALARY+(0.1*SALARY) AS "NEXT YEAR" FROM EMPLOYEES WHERE DEPTCODE='HDWRE';</pre>
Q13	Create a neatly formatted directory of all employees, including their department code and name. The list should be sorted first by department code, then by last name, then by first name.
	<pre>SELECT DEPTCODE, FIRSTNAME+' '+LASTNAME AS "NAME" FROM EMPLOYEES ORDER BY DEPTCODE ASC, LASTNAME ASC, FIRSTNAME ASC;</pre>

Note: Solutions provided are for your own reference and may have other possible variations or interpretations. In case of any query, kindly contact your lab instructors.