# LAB 3 SOLUTIONS

## Exercise 1

| | |
|---|---|
| **Q1** | Find the ID of the vendor who supplies grape. |
| | `SELECT VENDORID FROM VENDORS WHERE REPLNAME='Grape';` |
| **Q2** | Find all of the ingredients from the fruit food group with an inventory greater than 100. |
| | `SELECT NAME FROM INGREDIENTS WHERE FOODGROUP='Fruit' AND INVENTORY>100;` |
| **Q3** | Find the ingredients, unit price supplied by 'VGRUS' (vendor ID) order by unit price (asc). |
| | `SELECT NAME , UNITPRICE FROM INGREDIENTS WHERE VENDORID='VGRUS' ORDER BY UNITPRICE ASC;` |
| **Q4** | Find the date on which the last item was added. |
| | `SELECT MAX(DATEADDED) FROM ITEMS;` |
| **Q5** | Find the number of vendors each vendor referred, and only report the vendors referring more than one. |
| | `SELECT COUNT(VENDORID),REFERREDBY FROM VENDORS GROUP BY REFERREDBY HAVING COUNT(VENDORID)>1;` |

## Exercise 2

| | |
|---|---|
| **Q1** | Find the average salary for all employees. |
| | `SELECT AVG(SALARY) FROM EMPLOYEES;` |
| Q2 | Find the average salary of employees in every department. |
| | `SELECT DEPTCODE, AVG(SALARY) AS 'AVG SAL' FROM EMPLOYEES GROUP BY DEPTCODE;` |
| **Q3** | Find the minimum and maximum project revenue for all active projects that make money. |
| | `SELECT MIN(REVENUE), MAX(REVENUE) FROM PROJECTS WHERE (ENDDATE IS NULL OR ENDDATE> GETDATE()) AND REVENUE!=0;` |
| **Q4** | Find the number of projects that have been worked on or currently are being worked on by an employee. |
| | `SELECT COUNT(PROJECTID) FROM WORKSON WHERE EMPLOYEEID IS NOT NULL;` |
| **Q5** | Find the last name of the employee whose last name is last in dictionary order. |
| | `SELECT MAX(LASTNAME) FROM EMPLOYEES;` |
| **Q6** | Compute the employee salary standard deviation. As a reminder, the formula for the population standard deviation is as follows: $\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i-\bar{x})^2}$ |
| | `SELECT STDEV(SALARY) FROM EMPLOYEES;` |
| **Q7** | Find the number of employees who are assigned to some department. You may not use a WHERE clause. |
| | `SELECT COUNT(DEPTCODE) FROM EMPLOYEES;` |
| **Q8** | For each department, list the department code and the number of employees in the department. |
| | `SELECT COUNT(EMPLOYEEID), DEPTCODE FROM EMPLOYEES GROUP BY DEPTCODE ;` |
| **Q9** | For each department that has a project, list the department code and report the average revenue and count of all of its projects. |
| | `SELECT COUNT(PROJECTID),DEPTCODE,AVG(REVENUE) FROM PROJECTS GROUP BY DEPTCODE HAVING COUNT(DEPTCODE)>0;` |

| | |
|---|---|
| **Q10** | Find the employee ID of all employees where their assigned time to work on projects is 100% or more. |
| | `SELECT EMPLOYEEID FROM WORKSON GROUP BY EMPLOYEEID HAVING SUM(ASSIGNEDTIME*100)>=100;` |
| **Q11** | Calculate the salary cost for each department with employees that don't have a last name ending in "re" after giving everyone a 10% raise. |
| | `SELECT SUM(SALARY+(0.10*SALARY)), DEPTCODE FROM EMPLOYEES WHERE LASTNAME NOT LIKE '%re' GROUP BY DEPTCODE ;` |

## Exercise 3

| | |
|---|---|
| **Q1** | Find the names of all people who work in the Consulting department. Solve it two ways: 1) using only WHERE-based join (i.e., no INNER/OUTER/CROSS JOIN) 2) with CROSS JOIN. |
| | `SELECT DISTINCT(FIRSTNAME +' '+ LASTNAME) AS NAME FROM EMPLOYEES E,DEPARTMENTS D WHERE E.DEPTCODE= D.CODE AND D.NAME='Consulting';`<br><br>`SELECT DISTINCT(E.FIRSTNAME +' '+ E.LASTNAME) AS NAME FROM EMPLOYEES E CROSS JOIN DEPARTMENTS D WHERE E.DEPTCODE= D.CODE AND D.NAME='Consulting';` |
| **Q2** | Find the names of all people who work in the Consulting department and who spend more than 20% of their time on the project with ID ADT4MFIA. Solve three ways:<br>1) Using only WHERE-based join (i.e., no INNER/OUTER/CROSS JOIN), 2) using JOIN. |
| | `SELECT DISTINCT(FIRSTNAME +' '+ LASTNAME) AS NAME FROM EMPLOYEES E,DEPARTMENTS D , WORKSON W, (SELECT W.EMPLOYEEID,W.PROJECTID, (W.ASSIGNEDTIME/SATIME.SAT)*100 AS SAT FROM WORKSON W, (SELECT SUM(ASSIGNEDTIME) AS SAT, EMPLOYEEID FROM WORKSON GROUP BY EMPLOYEEID) SATIME WHERE W.EMPLOYEEID=SATIME.EMPLOYEEID) ASSIGN WHERE E.DEPTCODE= D.CODE AND D.NAME='Consulting' AND E.EMPLOYEEID = W.EMPLOYEEID AND W.PROJECTID='ADT4MFIA' AND ASSIGN.EMPLOYEEID=E.EMPLOYEEID AND ASSIGN.SAT>20;`<br><br>`SELECT DISTINCT(FIRSTNAME +' '+ LASTNAME) AS NAME FROM EMPLOYEES E JOIN DEPARTMENTS D ON E.DEPTCODE=D.CODE JOIN WORKSON W ON E.EMPLOYEEID = W.EMPLOYEEID JOIN (SELECT W.EMPLOYEEID,W.PROJECTID, (W.ASSIGNEDTIME/SATIME.SAT)*100 AS SAT FROM WORKSON W, (SELECT SUM(ASSIGNEDTIME) AS SAT, EMPLOYEEID FROM WORKSON GROUP BY EMPLOYEEID) SATIME WHERE W.EMPLOYEEID=SATIME.EMPLOYEEID) ASSIGN ON ASSIGN.EMPLOYEEID=E.EMPLOYEEID WHERE D.NAME='Consulting' AND W.PROJECTID='ADT4MFIA'AND ASSIGN.SAT>20;` |

| | |
|---|---|
| **Q3** | Find the total percentage of time assigned to employee Abe Advice. Solve it two ways: 1) using only WHERE-based join (i.e., no INNER/OUTER/CROSS JOIN) and 2) using some form of JOIN. |
| | `SELECT (SATIME.SAT/ TOTALASSIGN.TOTAL)*100 AS PERCENTAGE FROM (SELECT SUM(ASSIGNEDTIME)AS TOTAL FROM WORKSON) TOTALASSIGN, (SELECT SUM(ASSIGNEDTIME) AS SAT FROM WORKSON W, EMPLOYEES E WHERE W.EMPLOYEEID=E.EMPLOYEEID AND E.FIRSTNAME='Abe' AND E.LASTNAME='Advice' GROUP BY W.EMPLOYEEID) SATIME;`<br><br>`SELECT (SATIME.SAT/ TOTALASSIGN.TOTAL)*100 AS PERCENTAGE FROM (SELECT SUM(ASSIGNEDTIME)AS TOTAL FROM WORKSON) TOTALASSIGN, (SELECT SUM(ASSIGNEDTIME) AS SAT FROM WORKSON W JOIN EMPLOYEES E ON E.EMPLOYEEID = W.EMPLOYEEID WHERE E.FIRSTNAME='Abe' AND E.LASTNAME='Advice' GROUP BY W.EMPLOYEEID) SATIME;` |
| **Q4** | Find the descriptions of all projects that require more than 70% of an employee's time. Solve it two ways: 1) using only WHERE-based join (i.e., no INNER/OUTER/CROSS JOIN) and 2) using some form of JOIN. |
| | `SELECT DISTINCT DESCRIPTION FROM PROJECTS P, (SELECT PROJECTID, W.EMPLOYEEID, ((W.ASSIGNEDTIME/AST.ASSIGN)*100) AS AST1 FROM WORKSON W, (SELECT SUM(ASSIGNEDTIME) AS ASSIGN ,EMPLOYEEID FROM WORKSON GROUP BY EMPLOYEEID) AST WHERE W.EMPLOYEEID=AST.EMPLOYEEID) PRJ WHERE PRJ.PROJECTID=P.PROJECTID AND PRJ.AST1>70;`<br><br>`OR`<br><br>`SELECT DISTINCT DESCRIPTION FROM PROJECTS P JOIN (SELECT PROJECTID, W.EMPLOYEEID, ((W.ASSIGNEDTIME/AST.ASSIGN)*100) AS AST1 FROM WORKSON W JOIN (SELECT SUM(ASSIGNEDTIME) AS ASSIGN ,EMPLOYEEID FROM WORKSON GROUP BY EMPLOYEEID) AST ON W.EMPLOYEEID=AST.EMPLOYEEID) PRJ ON PRJ.PROJECTID=P.PROJECTID WHERE PRJ.AST1>70;` |
| **Q5** | For each employee, list the employee ID, number of projects, and the total percentage of time for the current projects to which she is assigned. Include employees not assigned to any project. |

| | |
|---|---|
| | SELECT E.EMPLOYEEID,<br>COUNT(W.PROJECTID) AS "NUMBER OF PROJECTS",<br>SUM(ASSIGNEDTIME)*100 AS "TOTAL PERCENTAGE OF TIME"<br>FROM EMPLOYEES E LEFT JOIN WORKSON W ON<br>E.EMPLOYEEID=W.EMPLOYEEID JOIN PROJECTS P ON |
| | W.PROJECTID=P.PROJECTID WHERE P.ENDDATE >GETDATE() OR<br>P.ENDDATE IS NULL GROUP BY E.EMPLOYEEID; |
| **Q6** | Find the description of all projects with no employees assigned to them. |
| | SELECT DISTINCT(DESCRIPTION) FROM PROJECTS P,WORKSON W WHERE<br>P.PROJECTID NOT IN (SELECT PROJECTID FROM WORKSON);<br><br>OR<br><br>SELECT DISTINCT(P.DESCRIPTION) FROM PROJECTS P LEFT JOIN<br>WORKSON W ON P.PROJECTID=W.PROJECTID WHERE W.EMPLOYEEID IS<br>NULL; |
| **Q7** | For each project, find the greatest percentage of time assigned to one employee. Solve it two ways: 1) using only WHERE-based join (i.e., no INNER/OUTER/CROSS JOIN) and 2) using some form of JOIN. |
| | SELECT PROJECTID, MAX(ASSIGNEDTIME) FROM EMPLOYEES E ,<br>WORKSON W WHERE E.EMPLOYEEID=W.EMPLOYEEID GROUP BY<br>W.PROJECTID;<br><br>SELECT PROJECTID, MAX(ASSIGNEDTIME) FROM EMPLOYEES E JOIN<br>WORKSON W ON E.EMPLOYEEID=W.EMPLOYEEID GROUP BY W.PROJECTID; |
| **Q8** | For each employee ID, find the last name of all employees making more money than that employee. Solve it two ways: 1) using only WHERE-based join (i.e., no INNER/OUTER/CROSS JOIN) and 2) using some form of JOIN. |
| | SELECT E1.EMPLOYEEID, E2.LASTNAME FROM EMPLOYEES E1,<br>EMPLOYEES E2 WHERE E2.SALARY>E1.SALARY;<br><br>SELECT E1.EMPLOYEEID, E2.LASTNAME FROM EMPLOYEES E1 CROSS<br>JOIN EMPLOYEES E2 WHERE E2.SALARY>E1.SALARY; |

## In-text Questions

| | |
|---|---|
| **Q1** | Rewrite the above query (Pg. 11) to use EXCEPT instead of an OUTER JOIN |

| | |
|---|---|
| | SELECT COMPANYNAME FROM VENDORS<br>EXCEPT<br>SELECT COMPANYNAME FROM VENDORS V,INGREDIENTS I WHERE V.VENDORID=I.VENDORID; |
| **Q2** | From set theory, R∩S = R-(R-S). Rewrite the following query using EXCEPT instead of INTERSECT (Pg. 11). |
| | SELECT ITEMID FROM MADEWITH M, INGREDIENTS I WHERE M.INGREDIENTID = I.INGREDIENTID AND FOODGROUP='Milk'<br>EXCEPT<br>(SELECT ITEMID FROM MADEWITH M, INGREDIENTS I WHERE M.INGREDIENTID = I.INGREDIENTID AND FOODGROUP='Milk'<br>EXCEPT<br>SELECT ITEMID FROM MADEWITH M, INGREDIENTS I WHERE M.INGREDIENTID = I.INGREDIENTID AND FOODGROUP='Fruit'); |

## Exercise 4

| | |
|---|---|
| **Q1** | Find all dates on which projects either started or ended. Eliminate any duplicate or NULL dates. Sort your results in descending order. |
| | SELECT DISTINCT(STARTDATE) AS DATES<br>FROM PROJECTS<br>WHERE STARTDATE IS NOT NULL<br>UNION<br>SELECT DISTINCT(ENDDATE) AS DATES FROM PROJECTS WHERE ENDDATE IS NOT NULL ORDER BY DATES DESC; |
| **Q2** | Display all the food groups from ingredients, in which 'grape' is not a member. |
| | SELECT FOODGROUP FROM INGREDIENTS<br>EXCEPT<br>SELECT FOODGROUP FROM INGREDIENTS WHERE NAME ='Grape' GROUP BY FOODGROUP HAVING FOODGROUP IS NOT NULL; |
| **Q2** | Use INTERSECT to find the first and last name of all employees who both work on the Robotic Spouse and for the Hardware department. |

| | |
|---|---|
| | SELECT FIRSTNAME, LASTNAME FROM EMPLOYEES INNER JOIN WORKSON ON EMPLOYEES.EMPLOYEEID=WORKSON.EMPLOYEEID INNER JOIN projects ps ON ps.projectid = workson.projectid AND ps.description ='Robotic Spouse' INTERSECT SELECT FIRSTNAME,LASTNAME FROM EMPLOYEES WHERE DEPTCODE='HDWRE'; |
| **Q3** | Use EXCEPT to find the first and last name of all employees who work on the Robotic Spouse but not for the Hardware department. |
| | SELECT FIRSTNAME, LASTNAME FROM EMPLOYEES INNER JOIN WORKSON ON EMPLOYEES.EMPLOYEEID=WORKSON.EMPLOYEEID INNER JOIN projects ps ON ps.projectid = workson.projectid AND ps.description ='Robotic Spouse' EXCEPT SELECT FIRSTNAME,LASTNAME FROM EMPLOYEES WHERE DEPTCODE='HDWRE'; |
| **Q4** | Find the first and last name of all employees who work on the Download Client project but not the Robotic Spouse project. |
| | SELECT FIRSTNAME, LASTNAME FROM PROJECTS P JOIN WORKSON W on P.PROJECTID = W.PROJECTID JOIN EMPLOYEES E ON W.EMPLOYEEID = E.EMPLOYEEID |

| | |
|---|---|
| | WHERE P.DESCRIPTION='Download Client' EXCEPT SELECT FIRSTNAME, LASTNAME FROM PROJECTS P JOIN WORKSON W on P.PROJECTID = W.PROJECTID JOIN EMPLOYEES E ON W.EMPLOYEEID =E.EMPLOYEEID WHERE P.DESCRIPTION='Robotic Spouse'; |
| **Q5** | Find the first and last name of all employees who work on the Download Client project and the Robotic Spouse project. |
| | SELECT FIRSTNAME, LASTNAME FROM PROJECTS P JOIN WORKSON W on P.PROJECTID = W.PROJECTID JOIN EMPLOYEES E ON W.EMPLOYEEID =E.EMPLOYEEID WHERE P.DESCRIPTION='Download Client' INTERSECT SELECT FIRSTNAME, LASTNAME FROM PROJECTS P JOIN WORKSON W on P.PROJECTID = W.PROJECTID JOIN EMPLOYEES E ON W.EMPLOYEEID =E.EMPLOYEEID WHERE P.DESCRIPTION='Robotic Spouse'; |

| | |
|---|---|
| **Q6** | Find the first and last name of all employees who work on either the Download Client project or the Robotic Spouse project. |
| | `SELECT FIRSTNAME, LASTNAME FROM PROJECTS P JOIN WORKSON W on P.PROJECTID = W.PROJECTID JOIN EMPLOYEES E ON W.EMPLOYEEID =E.EMPLOYEEID`<br>`WHERE P.DESCRIPTION='Download Client'`<br>`UNION`<br>`SELECT FIRSTNAME, LASTNAME FROM PROJECTS P JOIN WORKSON W on P.PROJECTID = W.PROJECTID JOIN EMPLOYEES E ON W.EMPLOYEEID =E.EMPLOYEEID WHERE P.DESCRIPTION='Robotic Spouse';` |
| **Q7** | Find the first and last name of all employees who work on either the Download Client project or the Robotic Spouse project but not both. |
| | `(SELECT FIRSTNAME, LASTNAME FROM PROJECTS P JOIN WORKSON W on P.PROJECTID = W.PROJECTID JOIN EMPLOYEES E ON W.EMPLOYEEID =E.EMPLOYEEID WHERE P.DESCRIPTION='Download Client'`<br>`UNION`<br>`SELECT FIRSTNAME, LASTNAME FROM PROJECTS P JOIN WORKSON W on P.PROJECTID = W.PROJECTID JOIN EMPLOYEES E ON W.EMPLOYEEID =E.EMPLOYEEID  WHERE P.DESCRIPTION='Robotic Spouse')`<br>`EXCEPT`<br>`(SELECT FIRSTNAME, LASTNAME FROM PROJECTS P JOIN WORKSON W on P.PROJECTID = W.PROJECTID JOIN EMPLOYEES E ON W.EMPLOYEEID =E.EMPLOYEEID WHERE P.DESCRIPTION='Download Client'`<br>`INTERSECT` |
| | `SELECT FIRSTNAME, LASTNAME FROM PROJECTS P JOIN WORKSON W on P.PROJECTID = W.PROJECTID JOIN EMPLOYEES E ON W.EMPLOYEEID =E.EMPLOYEEID  WHERE P.DESCRIPTION='Robotic Spouse');` |
| **Q8** | Using EXCEPT, find all of the departments without any projects. |
| | `SELECT NAME FROM DEPARTMENTS`<br>`EXCEPT`<br>`SELECT NAME FROM DEPARTMENTS D JOIN PROJECTS P ON D.CODE=P.DEPTCODE;` |

*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.*