EXERCISE 1:

1. Find the first and last name of all employees who work on the Download Client project but not on the Robotic Spouse project.

```
SELECT FIRSTNAME, LASTNAME
FROM PROJECTS P JOIN WORKSON W on P.PROJECTID = W.PROJECTID
JOIN EMPLOYEES E ON W.EMPLOYEEID =W.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
=W.EMPLOYEEID WHERE P.DESCRIPTION='Robotic Spouse';
```

2. Find the first and last name of all employees who work on the Download Client project and on the Robotic Spouse project

```
SELECT FIRSTNAME, LASTNAME FROM PROJECTS P JOIN WORKSON W on P.PROJECTID =
W.PROJECTID JOIN EMPLOYEES E ON W.EMPLOYEEID

=W.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
=W.EMPLOYEEID
WHERE P.DESCRIPTION='Robotic Spouse';
```

3. Find the first and last name of all employees who work on the Download Client project or on the Robotic Spouse project

```
SELECT FIRSTNAME, LASTNAME FROM PROJECTS P JOIN WORKSON W on P.PROJECTID =
W.PROJECTID JOIN EMPLOYEES E ON W.EMPLOYEEID
=W.EMPLOYEEID
WHERE P.DESCRIPTION='Download Client'
UNION
SELECT FIRSTNAME, LASTNAME FROM PROJECTS P JOIN WORKSON W on
P.PROJECTID = W.PROJECTID JOIN EMPLOYEES E ON W.EMPLOYEEID
=W.EMPLOYEEID WHERE P.DESCRIPTION='Robotic Spouse';
```

4. Find the names of all people who work in the Consulting department.

```
SELECT DISTINCT(E.FIRSTNAME +' '+ E.LASTNAME) AS NAME FROM
EMPLOYEES E CROSS JOIN DEPARTMENTS D WHERE E.DEPTCODE= D.CODE AND
D.NAME='Consulting';
```

5. 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.

```
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;
```

6. Find the total percentage of time assigned to employee Abe Advice.

```
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;
```

7. Find the descriptions of all projects that require more than 70% of an employee's time.

```
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;
```

8. 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,

COUNT(W.PROJECTID) AS "NUMBER OF PROJECTS",

SUM(ASSIGNEDTIME)*100 AS "TOTAL PERCENTAGE OF TIME"

FROM EMPLOYEES E LEFT JOIN WORKSON W ON

E.EMPLOYEEID=W.EMPLOYEEID JOIN PROJECTS P ON

W.PROJECTID=P.PROJECTID WHERE P.ENDDATE >GETDATE() OR P.ENDDATE IS NULL
GROUP BY E.EMPLOYEEID;
```

9. Find the description of all projects with no employees assigned to them.

```
SELECT DISTINCT(P.DESCRIPTION) FROM PROJECTS P LEFT JOIN WORKSON W ON
P.PROJECTID=W.PROJECTID WHERE W.EMPLOYEEID IS NULL;
```

10. For each project, find the greatest percentage of time assigned to one employee.

```
SELECT PROJECTID, MAX(ASSIGNEDTIME) FROM EMPLOYEES E JOIN
WORKSON W ON E.EMPLOYEEID=W.EMPLOYEEID GROUP BY W.PROJECTID;
```

11. For each employee ID, find the last name of all employees making more money than that employee.

```
SELECT E1.EMPLOYEEID, E2.LASTNAME FROM EMPLOYEES E1 CROSS JOIN EMPLOYEES
E2 WHERE E2.SALARY>E1.SALARY;
```

EXERCISES 2:

1. Create a view containing all of the employees assigned to the 'Robotic Spouse' project. Include the percent time they are assigned to the project.

```
1. CREATE VIEW EX1 AS
    SELECT W.EMPLOYEEID , E.FIRSTNAME , E.LASTNAME ,W.PROJECTID ,
        (W.ASSIGNEDTIME/(SELECT SUM(W2.ASSIGNEDTIME)
                FROM WORKSON W2 WHERE
                W.EMPLOYEEID=W2.EMPLOYEEID)) ASSIGNEDTIME
    FROM WORKSON W , EMPLOYEES E , PROJECTS P
    WHERE E.EMPLOYEEID = W.EMPLOYEEID AND W.PROJECTID = P.PROJECTID AND
    P.DESCRIPTION='Robotic Spouse';
```

2. Query your view created in the previous question to find the employee first and last name with the greatest amount of time assigned to 'Robotic Spouse'.

```
. SELECT
    FIRSTNAME,LASTNAME
    FROM EX1
    WHERE ASSIGNEDTIME=(SELECT MAX(ASSIGNEDTIME) FROM EX1);
```

3. Create a view of employees with their department name.

```
3. CREATE VIEW EDETAIL AS
    SELECT E.EMPLOYEEID, E.FIRSTNAME , E.LASTNAME , D.NAME, E.SALARY
    FROM EMPLOYEES E , DEPARTMENTS D WHERE E.DEPTCODE=D.CODE;
```

4. Query your view to find all the first and last names of employees in the Consulting department.

```
SELECT
    FIRSTNAME,LASTNAME
    FROM EDETAIL
    WHERE NAME='Consulting';
```

5. Create a view showing all of the projects assigned to Abe Advice, including his percentage time on each project.

```
5. CREATE VIEW AA1 AS
    SELECT W.PROJECTID, P.DEPTCODE, DESCRIPTION, STARTDATE, ENDDATE, REVENUE,
        (W.ASSIGNEDTIME/ (SELECT SUM (W2.ASSIGNEDTIME)
                          FROM WORKSON W2
                          WHERE W.EMPLOYEEID=W2.EMPLOYEEID)) ASSIGNEDTIME
    FROM EMPLOYEES E, WORKSON W, PROJECTS P
    WHERE E.EMPLOYEEID = W.EMPLOYEEID AND W.PROJECTID = P.PROJECTID AND
    FIRSTNAME='Abe' AND LASTNAME ='Advice';
```

6. Query your view to find the total amount of time Abe is assigned to projects.

```
SELECT SUM(ASSIGNEDTIME)
    FROM AA1;
```

7. Create an updatable view showing employees and their salaries. Give everyone a 10% raise by updating the view

```
7. CREATE VIEW OO AS
    SELECT EMPLOYEEID,FIRSTNAME,LASTNAME,SALARY
    FROM EMPLOYEES;
    UPDATE OO SET SALARY=(SALARY+(0.10*SALARY));
```