Using Functions, Subqueries, and ROLAP in SQL Queries

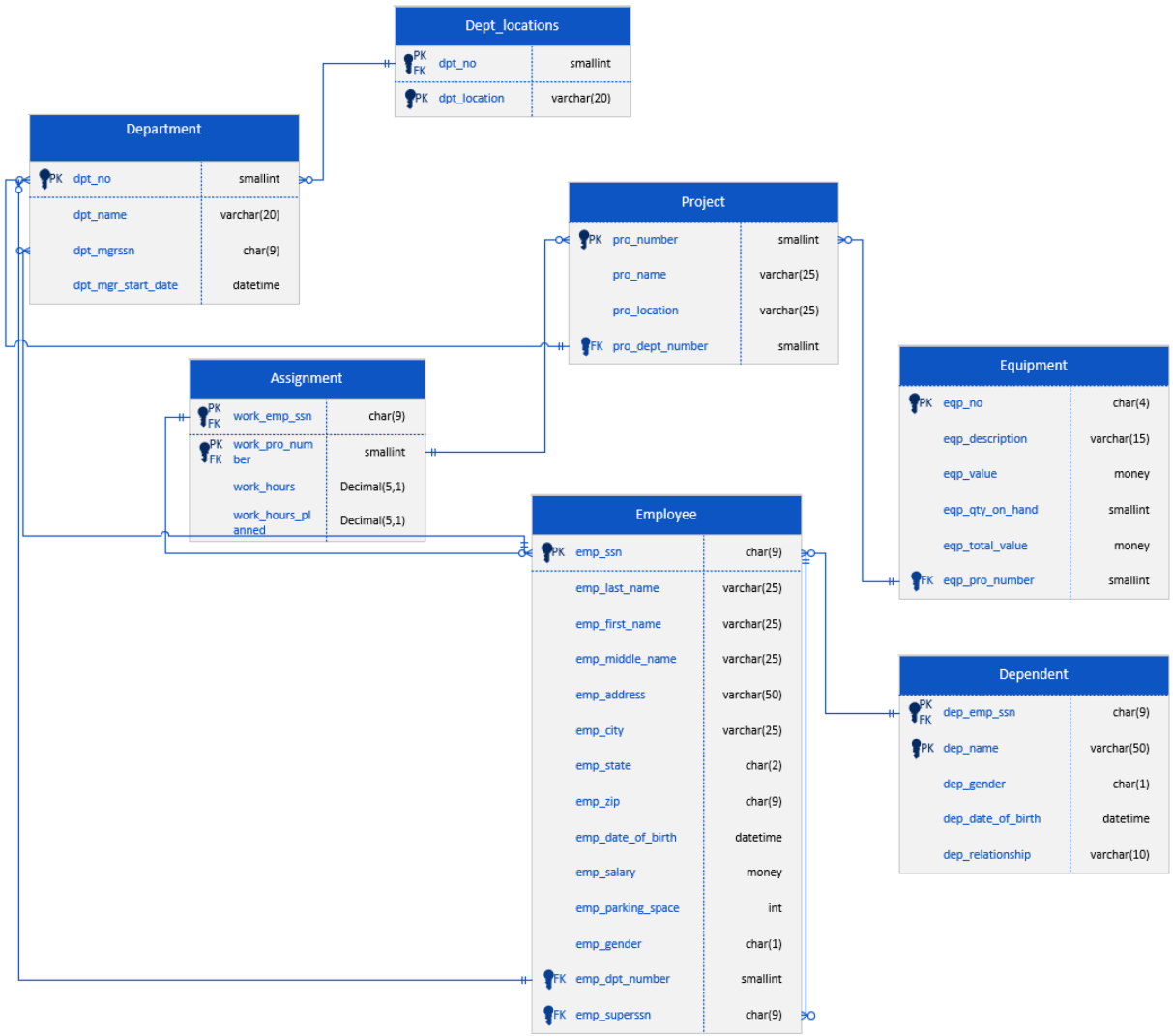
Physical Table Design of Sale_Co_DW.db, QC_Checks.db, or Company.db

This physical table structure for the three SQLite databases may be obtained via DBBrowser for SQLite or via .schema dot command in sqlite3.exe.

Sale_Co_DW.db is more of a data warehouse design having fact tables and dimension tables.

QC_Checks.db contains quality check errors from the case study reviewed during Week 05.

Company.db contains company data used to illustrate subqueries in Classwork 6.2. The logical model with attributes and relationships for Company.db follows:



Exercise:

1) Use any of the three databases above to complete each subsection in step 2. Do not use any of the subqueries illustrated in Classwork 6.1 or Classwork 6.2.

2) Create five unique, executable queries (2 points), using a minimum of four functions on average (2 points), having multiple grouping indexes (2 points), and :

   a. A Type I subquery (2 points) nested with two inner queries (2 points).
   b. A Type II subquery (2 points) nested with two inner queries (2 points).
   c. A Type III correlated subquery (2 points).
   d. The SELECT projection from a table created by a SELECT statement (2 points) with 5 columns.
   e. The SELECT projection from tables saved to a CSV file (2 points).

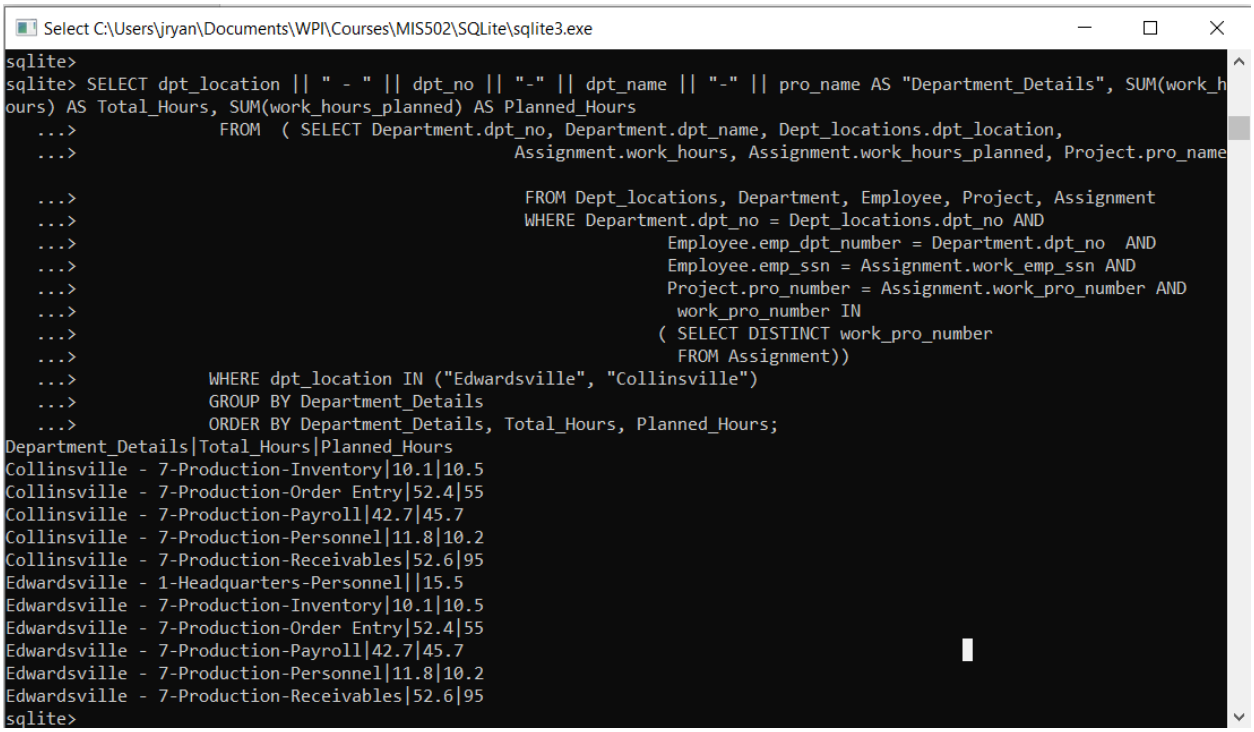
3) Submit to Canvas Assignment in one PDF document:

   a. Your SQL scripts for each query.
   b. Legible output projection from running each query.

Answers using Company.db:

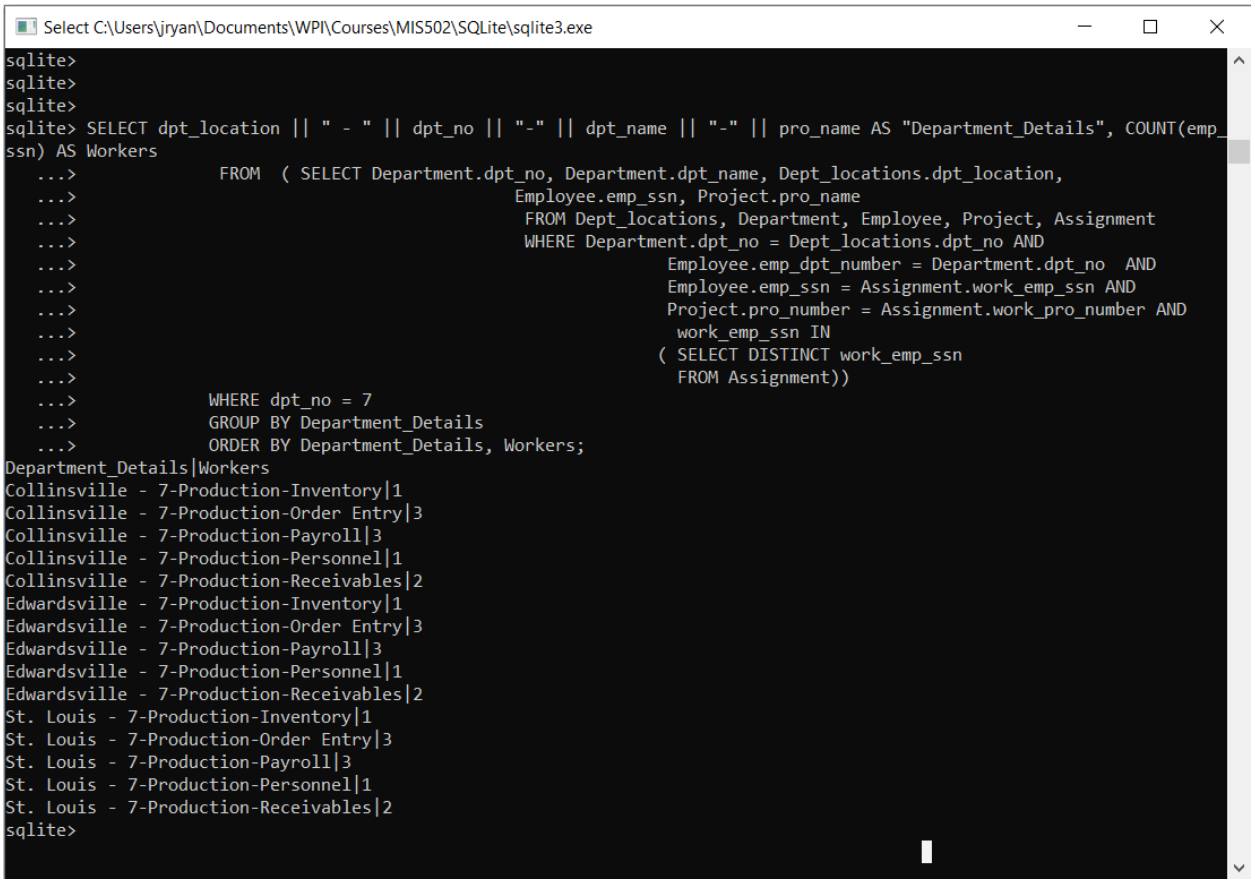
2a Type I subquery (2 points) nested with two inner queries (2 points).

```
SELECT dpt_location || " - " || dpt_no || "-" || dpt_name || "-" || pro_name AS "Department_Details",
SUM(work_hours) AS Total_Hours, SUM(work_hours_planned) AS Planned_Hours
        FROM  ( SELECT Department.dpt_no, Department.dpt_name, Dept_locations.dpt_location,
                        Assignment.work_hours, Assignment.work_hours_planned, Project.pro_name
                        FROM Dept_locations, Department, Employee, Project, Assignment
                        WHERE Department.dpt_no = Dept_locations.dpt_no AND
                                Employee.emp_dpt_number = Department.dpt_no  AND
                                Employee.emp_ssn = Assignment.work_emp_ssn AND
                                Project.pro_number = Assignment.work_pro_number AND
                                work_pro_number IN
                                ( SELECT DISTINCT work_pro_number
                                FROM Assignment))
        WHERE dpt_location IN ("Edwardsville", "Collinsville")
        GROUP BY Department_Details
        ORDER BY Department_Details, Total_Hours, Planned_Hours;
```



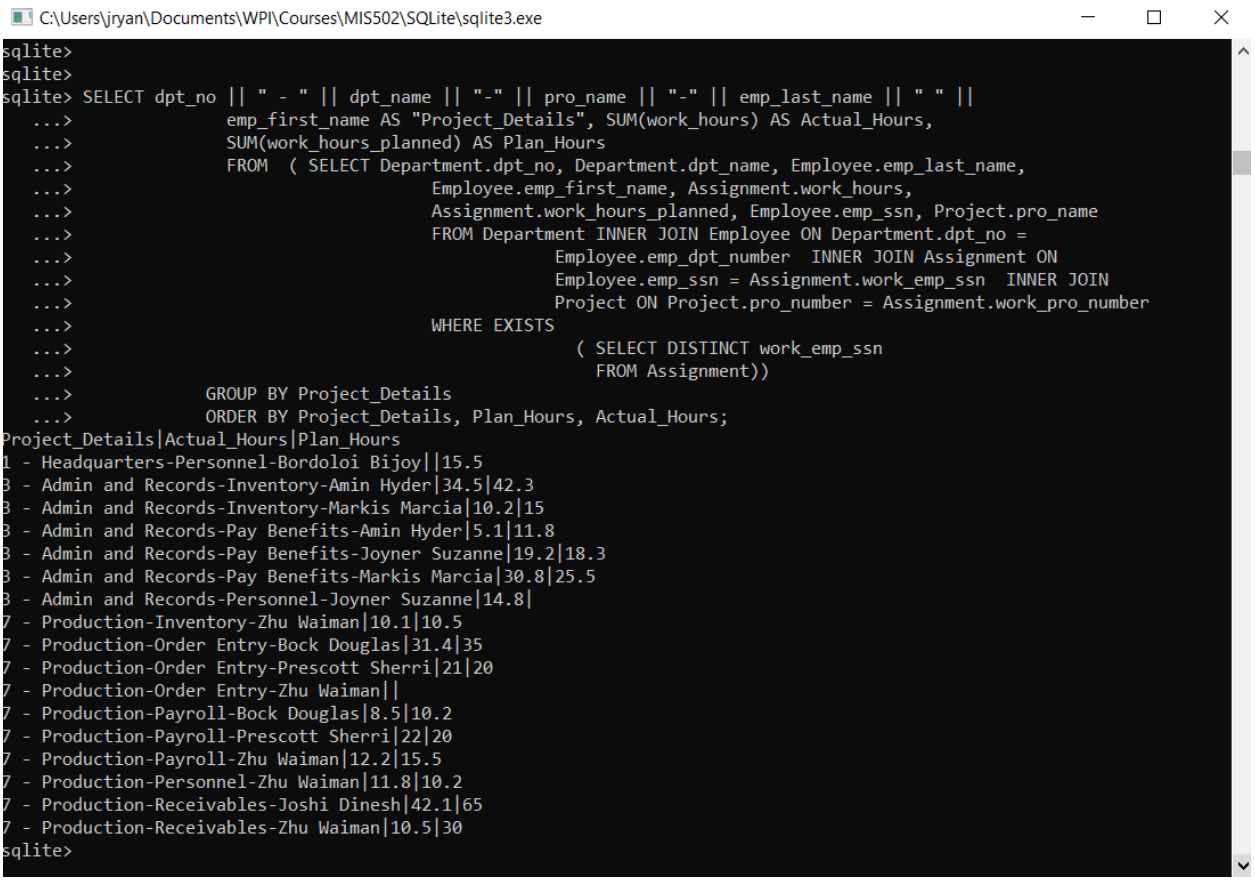
2b: A Type II subquery (2 points) nested with two inner queries (2 points).

```
SELECT dpt_location || " - " || dpt_no || "-" || dpt_name || "-" || pro_name AS "Department_Details",
COUNT(emp_ssn) AS Workers
        FROM  ( SELECT Department.dpt_no, Department.dpt_name, Dept_locations.dpt_location,
                        Employee.emp_ssn, Project.pro_name
                        FROM Dept_locations, Department, Employee, Project, Assignment
                        WHERE Department.dpt_no = Dept_locations.dpt_no AND
                                Employee.emp_dpt_number = Department.dpt_no  AND
                                Employee.emp_ssn = Assignment.work_emp_ssn AND
                                Project.pro_number = Assignment.work_pro_number AND
                                work_emp_ssn IN
                               ( SELECT DISTINCT work_emp_ssn
                                FROM Assignment))
        WHERE dpt_no = 7
        GROUP BY Department_Details
        ORDER BY Department_Details, Workers;
```



2c: A Type III correlated subquery (2 points).

```
SELECT dpt_no || " - " || dpt_name || "-" || pro_name || "-" || emp_last_name || " " ||
        emp_first_name AS "Project_Details", SUM(work_hours) AS Actual_Hours,
        SUM(work_hours_planned) AS Plan_Hours
        FROM  ( SELECT Department.dpt_no, Department.dpt_name, Employee.emp_last_name,
                Employee.emp_first_name, Assignment.work_hours,
                Assignment.work_hours_planned, Employee.emp_ssn, Project.pro_name
                FROM Department INNER JOIN Employee ON Department.dpt_no =
                        Employee.emp_dpt_number  INNER JOIN Assignment ON
                        Employee.emp_ssn = Assignment.work_emp_ssn  INNER JOIN
                        Project ON Project.pro_number = Assignment.work_pro_number
                WHERE EXISTS
                        ( SELECT DISTINCT work_emp_ssn
                          FROM Assignment))
        GROUP BY Project_Details
        ORDER BY Project_Details, Plan_Hours, Actual_Hours;
```



The correlated subquery above is an example of using EXISTS.

2d: A SELECT projection from a table created by a SELECT statement (2 points) with 5 columns.

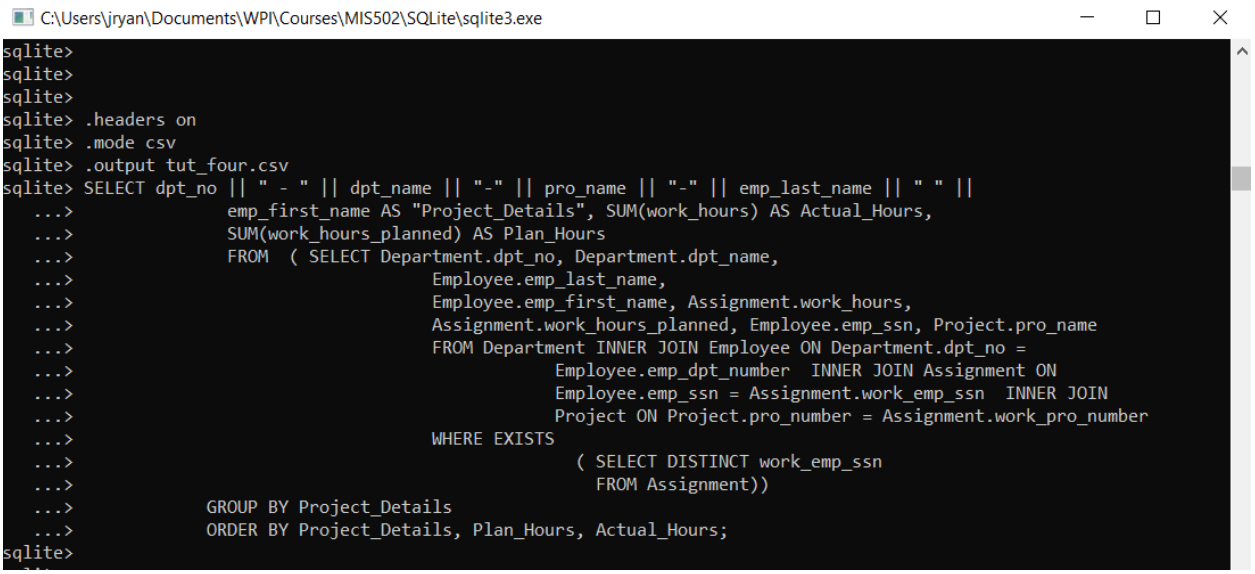
```
CREATE TABLE Production_Projects AS
SELECT dpt_no || " - " || dpt_name || "-" || pro_name || "-" || emp_last_name || " " ||
        emp_first_name AS "Project_Details", SUM(work_hours) AS Actual_Hours,
        SUM(work_hours_planned) AS Plan_Hours
        FROM  ( SELECT Department.dpt_no, Department.dpt_name,
                    Employee.emp_last_name,
                    Employee.emp_first_name, Assignment.work_hours,
                    Assignment.work_hours_planned, Employee.emp_ssn, Project.pro_name
                    FROM Department INNER JOIN Employee ON Department.dpt_no =
                        Employee.emp_dpt_number  INNER JOIN Assignment ON
                        Employee.emp_ssn = Assignment.work_emp_ssn  INNER JOIN
                        Project ON Project.pro_number = Assignment.work_pro_number
                    WHERE EXISTS
                        ( SELECT DISTINCT work_emp_ssn
                          FROM Assignment))
        GROUP BY Project_Details
        ORDER BY Project_Details, Plan_Hours, Actual_Hours;

SELECT * FROM Production_Projects;
```

```
C:\Users\jryan\Documents\WPI\Courses\MIS502\SQLite\sqlite3.exe
sqlite>
sqlite> CREATE TABLE Production_Projects AS
   ...> SELECT dpt_no || " - " || dpt_name || "-" || pro_name || "-" || emp_last_name || " " ||
   ...>              emp_first_name AS "Project_Details", SUM(work_hours) AS Actual_Hours,
   ...>              SUM(work_hours_planned) AS Plan_Hours
   ...>              FROM  ( SELECT Department.dpt_no, Department.dpt_name,
   ...>                                  Employee.emp_last_name,
   ...>                                  Employee.emp_first_name, Assignment.work_hours,
   ...>                                  Assignment.work_hours_planned, Employee.emp_ssn, Project.pro_name
   ...>                                  FROM Department INNER JOIN Employee ON Department.dpt_no =
   ...>                                            Employee.emp_dpt_number  INNER JOIN Assignment ON
   ...>                                            Employee.emp_ssn = Assignment.work_emp_ssn  INNER JOIN
   ...>                                            Project ON Project.pro_number = Assignment.work_pro_number
   ...>                                  WHERE EXISTS
   ...>                                              ( SELECT DISTINCT work_emp_ssn
   ...>                                                FROM Assignment))
   ...>            GROUP BY Project_Details
   ...>            ORDER BY Project_Details, Plan_Hours, Actual_Hours;
sqlite>
sqlite>
sqlite> SELECT * FROM Production_Projects;
Project_Details|Actual_Hours|Plan_Hours
1 - Headquarters-Personnel-Bordoloi Bijoy||15.5
3 - Admin and Records-Inventory-Amin Hyder|34.5|42.3
3 - Admin and Records-Inventory-Markis Marcia|10.2|15
3 - Admin and Records-Pay Benefits-Amin Hyder|5.1|11.8
3 - Admin and Records-Pay Benefits-Joyner Suzanne|19.2|18.3
3 - Admin and Records-Pay Benefits-Markis Marcia|30.8|25.5
3 - Admin and Records-Personnel-Joyner Suzanne|14.8|
7 - Production-Inventory-Zhu Waiman|10.1|10.5
7 - Production-Order Entry-Bock Douglas|31.4|35
7 - Production-Order Entry-Prescott Sherri|21|20
7 - Production-Order Entry-Zhu Waiman||
7 - Production-Payroll-Bock Douglas|8.5|10.2
7 - Production-Payroll-Prescott Sherri|22|20
7 - Production-Payroll-Zhu Waiman|12.2|15.5
7 - Production-Personnel-Zhu Waiman|11.8|10.2
7 - Production-Receivables-Joshi Dinesh|42.1|65
7 - Production-Receivables-Zhu Waiman|10.5|30
sqlite>
```

2e: The SELECT projection from tables saved to a CSV file (2 points).

```
.headers on
.mode csv
.output tut_four.csv
SELECT dpt_no || " - " || dpt_name || "-" || pro_name || "-" || emp_last_name || " " ||
        emp_first_name AS "Project_Details", SUM(work_hours) AS Actual_Hours,
        SUM(work_hours_planned) AS Plan_Hours
        FROM  ( SELECT Department.dpt_no, Department.dpt_name,
                Employee.emp_last_name,
                Employee.emp_first_name, Assignment.work_hours,
                Assignment.work_hours_planned, Employee.emp_ssn, Project.pro_name
                FROM Department INNER JOIN Employee ON Department.dpt_no =
                        Employee.emp_dpt_number  INNER JOIN Assignment ON
                        Employee.emp_ssn = Assignment.work_emp_ssn  INNER JOIN
                        Project ON Project.pro_number = Assignment.work_pro_number
                WHERE EXISTS
                        ( SELECT DISTINCT work_emp_ssn
                          FROM Assignment))
        GROUP BY Project_Details
        ORDER BY Project_Details, Plan_Hours, Actual_Hours;
```



Use Notepad or another text editor to view the output file tut_four.csv.