8.1 A database can be created using existing data (from things like spreadsheets or tables), for a new systems development project, or for a database redesign

8.2 Database redesign is necessary because data is always changing. The database may need to be updated with new information(sometimes in bulk) or may need old information that is no longer needed deleted. Sometimes the database maybe large and complicated and may need several steps of redesign to complete.

8.3 New information has the capability to change the behavior in an organization; let’s say that new data provided shows that there are customers are more willing to buy a bag that contains more green candies than yellow candies. The company will adjust how many green candies come in one bag in order to sell more bags of candy. The company may decide, based on the previous information, to add a new product that contains only green candies. These changes may lead to redesigning the database to include information on the product (such as sales and customer information).

8.4 In order to determine if AdviserPhone determines the AdviserName we need to check if the information holds up using the EXIST and NOT EXIST correlated subqueries in SQL. These operators test whether or not there are any values returned by the subquery. That will indicate whether or not the values are meeting the conditions of the subquery.

8.5 /\* \*\*\*SQL-Noncorrelated-Subquery\*\*\* \*/

SELECT AvailableDrFirstName AvailableDrLastName

FROM DR AVAILABLE

WHERE AvailableDrID IN

(SELECT DepartmentID

FROM HOSPITAL DEPARTMENTS

WHERE Dep.Name = ‘Cardiology’) ;

8.6 We use nested statements in a correlated subquery because a DBMS cannot run the 2nd SELECT by itself or first as it depends on the 1st SELECT in order to run the query. It cannot be read from the bottom up like a regular subquery.

WHERE \_\_\_\_=\_\_\_\_\_

AND \_\_\_\_\_<>\_\_\_\_\_

These are used to compare the values of the two sets (the first SELECT and the second SELECT). It cannot process the subquery portion independent of the first SELECT.

8.7 /\* \*\*\*SQL-Correlated-Subquery\*\*\* \*/

SELECT D1.DrHospitalID, D1.Department, D1.DeprtName

FROM DRS EMPLOYEED AS D1

WHERE EXIST

(SELECT D2.Department

FROM DRS EMPLOYEED AS E2

WHERE D1.Department= D2.Department

AND D1.DeprtName <> D2.DeprtName) ;

8.8 Some of the notable differences are that in WHERE we use the operator EXIST instead of writing where the information is to be found. Then in our nested subquery we begin to compare the values from the original table (D1) to the copied tables (D2).

8.9 If we do not add in the operator EXIST we will end up with an empty set without any data and we will not be able to check for any functional dependencies. The correlated dependency on page 400 will always yield an empty data set.

8.10 /\* \*\*\* SQL- Correlated-Subquery\*\*\* \*/

SELECT A1.AdvisorPhone, A1.AdvisorName

FROM ADVISOR AS A1

WHERE A1.AdvisorPhone IN

(SELECT A2.AdvisorPhone

FROM ADVISOR AS A2

WHERE A1.AdvisorPhone = A2.AdvisorPhone

AND A1.AdvisorName <> A2.AdvisorName) ;

8.11 EXIST is a comparison operator that tests whether or not there are any values returned by the subquery. If values are returned, that indicates that the values meet the conditions of the subquery. So if the values for A1.AdvisorPhone correlate with A2.AdvisorPhone then non-empty values are returned. The same happens for A1.AdvisorName and A2.AdvisorName.

8.12 /\* \*\*\* SQL-Correlated-Subquery\*\*\* \*/

SELECT A1.AdvisorPhone, A1.AdvisorName

FROM ADVISOR AS A1

WHERE EXISTS

(SELECT A2.AdvisorPhone

FROM ADVISOR AS A2

WHERE A1.AdvisorPhone = A2.AdvisorPhone

AND A1.AdvisorName <> A2.AdvisorName) ;

8.13 The operator EXIST will bring up any values returned by the subquery. If there are that means that those values meet the conditions of the subquery. The operator NOT EXIST will return and empty set only if all rows in the subquery fail to meet the conditition. This can be used to find rows that do not match a condition.

8.14 A regular subquery can be processed from the bottom up, this means that results from the lowest query can be determined and then used to evaluate the upper-level queries. A correlated subquery has a statement that is nested within the main query. This is because the nested query depends on the main query (the first SELECT) in order to retrieve information. It cannot run from the bottom up.