Assignment 3 Solution:

1. Exercise 5.9 (15 points)

5.9 Write an extended relational-algebra view equivalent to the Datalog rule

\[ p(A, C, D) \leftarrow q(A, B), \ q2(B, C), \ q3(A, B), \ D = B + 1. \]

**Answer:** Let us assume that \(q1, q2\) and \(q3\) are instances of the schema \((A1, A2)\).

The relational algebra view is

\[
\text{create view } P \text{ as } \\
\Pi_{q1.A1,q2.A2,q3.A2+1}((\tau_{q3.A1=A1} \land q1.A2=q2.A1 \land q3.A2=q3.A2)(q1 \times q2 \times q3))
\]

2. Exercise 6.3 (Hint: you can create a syntax extension based on SQL and suppose SQL support your new clause, then explain how the system works for your extension) (15 points)

6.3 Referential-integrity constraints as defined in this chapter involve exactly two relations. Consider a database that includes the following relations:

- **salaried-worker** (name, office, phone, salary)
- **hourly-worker** (name, hourly-wage)
- **address** (name, street, city)

Suppose that we wish to require that every name that appears in **address** appear in either **salaried-worker** or **hourly-worker**, but not necessarily in both.

**a.** Propose a syntax for expressing such constraints.

**b.** Discuss the actions that the system must take to enforce a constraint of this form.

**Answer:**

**a.** For simplicity, we present a variant of the SQL syntax. As part of the **create table** expression for **address** we include

**foreign key** (name) **references** salaried-worker or hourly-worker

**b.** To enforce this constraint, whenever a tuple is inserted into the **address** relation, a lookup on the **name** value must be made on the **salaried-worker** relation and (if that lookup failed) on the **hourly-worker** relation (or vice-versa).
3. Exercise 7.9. (10 points)

7.9 Use Armstrong’s axioms to prove the soundness of the decomposition rule.

Answer: The decomposition rule, and its derivation from Armstrong’s axioms are given below:

- if \( \alpha \rightarrow \beta \gamma \), then \( \alpha \rightarrow \beta \) and \( \alpha \rightarrow \gamma \).
- \( \alpha \rightarrow \beta \gamma \) given
- \( \beta \gamma \rightarrow \beta \) reflexivity rule
- \( \alpha \rightarrow \beta \) transitivity rule
- \( \beta \gamma \rightarrow \gamma \) reflexive rule
- \( \alpha \rightarrow \gamma \) transitive rule

4. Exercise 7.12. (10 points)

7.12 Using the functional dependencies of Exercise 7.11, compute \( B^+ \).

Answer: Computing \( B^+ \) by the algorithm in Figure 7.7 we start with result = \{B\}. Considering FDs of the form \( \beta \rightarrow \gamma \) in \( F \), we find that the only dependencies satisfying \( \beta \subseteq \text{result} \) are \( B \rightarrow B \) and \( B \rightarrow D \). Therefore result = \{B, D\}. No more dependencies in \( F \) apply now. Therefore \( B^+ = \{B, D\} \).

5. Exercise 7.21 & 7.24 (25 points)

7.21 Give a lossless-join decomposition into BCNF of schema \( R \) of Exercise 7.2.

Answer: From Exercise 7.11, we know that \( B \rightarrow D \) is nontrivial and the left hand side is not a superkey. By the algorithm of Figure 7.13 we derive the relations \{\( \{A, B, C, E\}, \{B, D\}\)\}. This is in BCNF.

7.24 Give a lossless-join, dependency-preserving decomposition into 3NF of schema \( R \) of Exercise 7.2.

Answer: First we note that the dependencies given in Exercise 7.2 form a canonical cover. Generating the schema from the algorithm of Figure 7.14 we get

\[ R' = \{(A, B, C), (C, D, E), (B, D), (E, A)\} \]

Schema \( (A, B, C) \) contains a candidate key. Therefore \( R' \) is a third normal form dependency-preserving lossless-join decomposition.

Note that the original schema \( R = (A, B, C, D, E) \) is already in 3NF. Thus, it was not necessary to apply the algorithm as we have done above. The single original schema is trivially a lossless join, dependency-preserving decomposition.
6. Consider the following relation about published books:

\[
PUBLISHED\_BOOK\ (\ Title,\ Author,\ Book\_type,\ Price,\ Author\_affil,\ Publisher )
\]

where Author_affil indicates author's affiliation. And suppose there are dependencies as follows:

\{
\text{Title} \rightarrow \text{Book\_type, Publisher}
\}

\text{Book\_type} \rightarrow \text{Price}

\text{Author} \rightarrow \text{Author\_affil}\}

1. Identify the best normal form that the relation satisfy (1NF, 2NF (see the definition in Exercise 7.26), 3NF, BCNF, or 4NF). Why?

2. If it is not in BCNF, decompose it into a set of BCNF relations. Explain your decomposition.

Answer:

a) It is in 1NF but not 2NF, due to partial dependencies, such as Author\rightarrow Author\_affil.

b) A BCNF decomposition yields the following four relations

- \text{Book\_Type, Price}
- \text{Title, Book\_type, Publisher}
- \text{author, author\_affil}
- \text{Title, author} (a common mistake was to forget this relation)