1. What is the result of the following boolean expression when \((x, y, z)\) is as specified below? [3 pts]

\[
( (x = 10) \text{ OR } (y \text{ IS NOT NULL}) ) \text{ AND } (z > 20)
\]

- \((x = 10, y = 100, z = 30)\): **Answer: true**
- \((NULL, 100, 10)\): **Answer: false**
- \((NULL, NULL, 10)\): **Answer: false**
- \((15, 100, NULL)\): **Answer: unknown**
- \((10, NULL, 30)\): **Answer: true**
- \((NULL, NULL, NULL)\): **Answer: unknown**

2. For the relation schema: \(\text{student}(\text{name}, \_\_\_\_, \text{majorcode}, \text{gpa})\), where \(\_\_\_\_\_\) has been defined to be the primary key, the query:

\[
\text{select } \_\_\_\_, \text{name} \text{ from student group by } \_\_\_\_
\]

is invalid (will return an error), even though it seems to make sense. Why? How would you fix the query without changing the output? [3 pts]

**Answer:** The attributes in select clause must be either aggregates or must be present in the group by clause.

\[
\text{select } \_\_\_\_, \text{name} \text{ from student group by } \_\_\_\_, \text{name};
\]

3. I need to preserve dependencies, and a BCNF decomposition is not possible. Which of the following normal forms are still attainable (while preserving dependencies)? Circle your answers. [3 pts]

1NF 3NF 4NF

**Answer:** 1NF, 3NF are always attainable, even while needing to preserve dependencies. Since 4NF implies BCNF, 4NF is not attainable in this case.
4. Consider two relations \( R(A, B) \) and \( S(B, C) \) and a view on them defined as: \( V(A, B, C) = R \bowtie S \). Under what condition(s) can we allow a tuple insertion into \( V \), and what would its effect be? [3 pts]

**Answer:** If \( B \) is a candidate key for both relations, then we can safely allow inserts into the view. Inserting \((a, b, c)\) into \( V \) would result in inserts of \((a, b)\) and \((b, c)\) into the \( R \) and \( S \) respectively.

5. Given a relational schema over three attributes, \( A, B, \) and \( C \), that has three functional dependencies \( A \rightarrow B, B \rightarrow C, \) and \( A \rightarrow C \), both the decompositions \( (AB, BC) \) and \( (AB, AC) \) are in BCNF. Which one of them is preferable? Why? [3 pts]

**Answer:** The former is dependency-preserving; latter is not.

6. What does the *unknown* keyword/functionality mean? Briefly explain why it is needed. [3 pts]

**Answer:** It is used to indicate that the result of a boolean expression can not be evaluated. It is primarily used for dealing with NULLs.

7. Explain what *on delete cascade* means in the context of referential integrity constraints. [3 pts]

**Answer:** It means that when a tuple is deleted from a relation, any tuples that refer to it in some other relation should also be deleted.

8. What is a *prime* attribute? Where is it used in normalization? [3 pts]

**Answer:** A prime attribute is defined to be an attribute that is contained in some candidate key for the relation. It is used to define the 3rd normal form (3NF).

9. Given a relational schema \( R(A, B, C, D, E) \), with functional dependencies \( AB \rightarrow C, E \rightarrow A, \) and \( C \rightarrow E \), is the decomposition into \( R1(A, B, C) \) and \( R2(C, D, E) \) “lossless”? Why/why not? [3 pts]

**Answer:** It is not lossless. The intersection of the two schemas, \( R1 \cap R2 \) is \( C \), and \( C^+ = ACE \), so it is not a superkey for either \( R1 \) or \( R2 \).

10. The figure shows three different ways to model the entities *person*, *actor*, and *director*. Briefly explain what each one is trying to do, what issues you might face in using them, and which one would be the best option among these. [6 pts]

   ![Diagram](image)

   **Answer:** The second option is the best. The first option is incorrect as the relationship between Actor/Director and Person is not a “belongs to” relationship. The third option leads to wasted space.
especially when there are people who both direct, and act. The information about persons is also all over the place, making querying difficult.

11. Construct the relational schemas corresponding to the first two E/R models ((i) and (ii)) from above (irrespective of whether they make sense). Underline the primary keys. [4 pts]
   **Answer:** (i) Director(Trademark-Scene, SSN); Actor(Acting-School, SSN); Person(SSN, Name);
   (ii) Director(Trademark-Scene, SSN, Name); Actor(Acting-School, SSN, Name); Person(SSN, Name);

12. **Antijoin** is a binary operator similar to a join. Given two relations, \( R(A, B) \) and \( S(B, C) \), the result of the antijoin operation \( R \bowtie S \) has the same schema as \( R \), and contains all tuples of \( R \) that do not have at least one match (on attribute \( B \)) in \( S \). For example, if \( R(A, B) = \{(1, \alpha), (2, \beta), (3, \gamma)\} \) and \( S(B, C) = \{(\alpha, 2), (\gamma, 3)\} \), then \( R \bowtie S = \{(2, \beta)\} \). Express \( R \bowtie S \) using the basic relational operators (\( \sigma, \pi, \times, \cup, - \)). [5 pts]
   **Answer:**
   \[
   R \bowtie S = R - \pi_{A,B}(R \bowtie_{R,B=S,B} S)
   \]

13. Given the relation schema: \( R(A, B, C, D) \), and FDs on it: [10 pts]
    \( AB \to C, C \to D, D \to B. \)
    - List all candidate keys.
    - Is the relation in BCNF? List one FD that violates it if it is not.
    - Decompose the relation into BCNF if it is not already in it.
    - Is your decomposition into BCNF dependency-preserving?
    - Is the relation (\( R \)) in 3NF? List one FD that violates it if it is not.

   **Answer:**
   - \( AD, AB, AC \). Reason: \( A \) must be in any candidate key. All three possibilities after that work.
   - No. e.g. \( C \to D \) violates it.
   - (a) Decompose using \( C \to D \). Gives us CD and CAB.
   - (b) CAB is not in BCNF (\( C \to B \), inferred from the last two). Decompose it using \( C \to B \), to get CB, CA (and CD from above).
   - No. Unlikely there exists a dependency preserving decomposition.
   - Yes. All attributes are prime.
The following questions are to be answered on the following simplified *olympics* relational schema, where only the individual events information is stored for one *olympics* (say Athens 2004).

- Players (player-id, name, countryname, age);
- Events (event-id, name, eventtype);
- Results (player-id, event-id, medal);

*eventtype* can take values: *SWI* (swimming), *ATH* (athletics), *GYM* (gymnastics), etc.
*medal* can take values: *gold*, *silver*, *bronze*.
*age* is an integer. Assume player names are unique.

14. Write a relational algebra expression to find the names of the players who won at least one gold and one silver. [3 pts]

   **Answer:** \( \pi_{\text{name}} ((\pi_{\text{player-id}} (\sigma_{\text{medal} = ' \text{gold}' } (\text{results}))) \cap \pi_{\text{player-id}} (\sigma_{\text{medal} = ' \text{silver}' } (\text{results}))) \bowtie \text{Players} ) \)

15. Explain what the following expression does. [3 pts]

   \[ \pi_{\text{name}} ((\pi_{\text{player-id}} (\text{players}) - \pi_{\text{player-id}} (\text{results})) \bowtie \text{Players} ) \]

   **Answer:** It finds the players who did not win a medal.

16. Explain what the following expression does. [3 pts]

   \[ \pi_{\text{countryname}, \text{eventtype}} (\sigma_{\text{medal} = ' \text{gold}' } (\text{players} \bowtie \text{results} \bowtie \text{events} )) \div \pi_{\text{eventtype}} (\text{events} ) \]

   **Answer:** It finds the countries who won at least one gold in each of the event types.

17. Write a relational algebra query to find the names of all the players with the minimum age. Assume player names are unique. [5 pts]

   **Answer:**
   \[ \pi_{\text{name}} (\text{players}) - \pi_{p1.\text{name}} (\rho_{p1.\text{age} > p2.\text{age}} \rho_{p2} (\text{players})) \]

   The second expression (after -) finds all players such that there is at least one other player younger than them. We then just subtract it from all players to find the ones with minimum age.
The following questions are to be answered on the following simplified friends relational schema.

Person (name, address);
Friends (name1, name2);

where the second relation stores the information about friends (i.e., name1 and name2 are friends). Assume that this is symmetric relationship, and if X and Y are friends, then the second relation contains two tuples (X, Y), and (Y, X).

18. What does the following relational algebra expression do? Specifically, precisely list all types of tuples that will appear in the answer. [3 pts]

\[ \pi_{name1, name2}(\rho_{f1.name1 = f2.name2}(friends)) \]

**Answer:** For each person X, the result will contain a tuple (X, X), and tuples (X, Z) such that X is a friend of a friend of Z.

19. If the friends relation contains a tuple (X, Y), under what conditions would the above expression also contain that tuple? [3 pts]

**Answer:** Only if X had another friend Z who was also friends with Y.

20. How would you change the data in the relations so that the result of the above expression always contains tuple (X, Y) if the friends relation contained tuple (X, Y). [5 pts]

**Answer:** Add the tuple (X, X) to the friends relation for all X.

21. Rewrite the following SQL query so that it does not use a subquery. [3 pts]

```sql
select name
from person p
where exists (select *
from friends r, person p2
where r.name1 = p.name and p2.name = r.name2 and p.address = p2.address)
```

**Answer:**

```sql
select distinct name
from person p, friends r, person p2
where r.name1 = p.name and p2.name = r.name2 and p.address = p2.address
```

The `distinct` is very important.