Assignment 3

Question 2 (4 marks)

Ideally, $2^3 = 8$. However, none of the 9 intersections distinguishes the 8 relations evenly. Therefore, choice of 3 cannot be sufficient. A minimum of size 4 has to be chosen:

$$
\Gamma_9(A,B) = \begin{pmatrix}
A^* \cap B^* & A^* \cap \partial B & A^* \cap B^-\\
\partial A \cap B^* & \partial A \cap \partial B & \partial A \cap B^-\\
A^{-} \cap B^* & A^{-} \cap \partial B & A^{-} \cap B^-
\end{pmatrix}
$$

Here is a complete list of all correct answers:

(1 1 0) (1 0 1) (0 1 1) (1 1 1) (1 1 0) (1 1 0) (1 0 1) (0 1 0) (0 1 0)
(0 0 0) (0 1 0) (0 0 0) (0 0 0) (0 0 0) (0 0 0) (0 1 0) (1 0 0) (1 0 0)
(0 1 1) (0 1 0) (0 0 1) (1 0 0) (0 1 0) (0 0 0) (0 0 0) (1 1 0) (1 1 0)
(0 1 0) (1 1 0) (0 1 1) (0 1 1) (0 1 1) (0 1 1) (0 0 0) (1 0 0) (1 0 0)
(0 1 0) (0 1 0) (1 1 0) (1 1 0) (0 1 0) (0 1 0) (0 0 0) (1 1 0) (1 1 0)
(0 1 0) (0 1 0) (0 1 0) (0 1 0) (0 1 0) (0 1 0) (0 1 0) (0 1 0) (1 1 0)

And if you’re interested, here is a piece of Java code that finds out all the answers:

```java
import java.util.HashSet;

public class BF {
    private static String[] table = {
        "001001111", "111001001", "100100111", "100010001", "001011111", "111011001", "111011101", "110110111", "111111111"
    };
    public static void main(String[] args) {
        int min = 10;
        for (int mask = 0; mask < 1 << 9; mask++) {
            HashSet<String> set = new HashSet<String>();
            for (int i = 0; i < 8; i++) {
                String s = "";
                for (int j = 0; j < 9; j++)
                    if (((mask & (1 << j)) > 0))
                        s += table[i].charAt(j);
                set.add(s);
            }
            if (set.size() == 8 && Integer.bitCount(mask) <= min) {
                min = Integer.bitCount(mask);
                System.out.println(min);
                for (int i = 0; i < 3; i++) System.out.println()
                    for (int j = 0; j < 3; j++)
                        System.out.print((mask & (1 << (i * 3 + j))) > 0 ? 1 : 0);
            }
        }
    }
}
```
Question 3 (8 marks)

1) (2 marks)

```sql
SELECT R.NAME
FROM Road R, Building B
WHERE Cross( R.GEOMETRY, B.GEOMETRY ) = 1 AND B.NAME = 'Computer Science and Engineering';
```

2) (2 marks)

```sql
SELECT B.NAME
FROM HelpPoint H, Building B
WHERE Contains( Buffer(H.GEOMETRY, 1), B.GEOMETRY ) = 1 AND H.CODE = '001';
```

```sql
or
SELECT B.NAME
FROM HelpPoint H, Building B
WHERE Within(B.GEOMETRY, Buffer(H.GEOMETRY, 1)) = 1 AND H.CODE = '001';
```

```sql
or
SELECT B.NAME
FROM HelpPoint H, Building B
WHERE Equal( Intersection(B.GEOMETRY, Buffer(H.GEOMETRY, 1)), B.GEOMETRY ) AND H.CODE = '001';
```

3) (2 marks)

```sql
SELECT B.NAME
FROM Building B
WHERE NOT EXISTS( SELECT B1.NAME

```
```sql
FROM Building B1
WHERE Touch( B.GEOMETRY, B1.GEOMETRY )
```
```sql
);
```

Note: This is a sample solution assuming that the interiors of any two buildings do not intersect. We also accept other valid interpretations and assumptions.
4) (2 marks)

\[
\text{SELECT H1.CODE} \\
\text{FROM Building B1, HelpPoint H1} \\
\text{WHERE B1.NAME='Computer Science and Engineering' AND Distance(H1.GEOMETRY, B1.GEOMETRY) = ALL (SELECT Distance(H2.GEOMETRY, B1.GEOMETRY) FROM HelpPoint H2)} \\
\]