1)

a)

Boyce-Codd Normal form (BCNF):

$\alpha \rightarrow \beta$

where $\alpha \subseteq R$ and $\beta \subseteq R$, at least one of the following holds:

$\alpha \rightarrow \beta$ is trivial

α is a superkey for R

Steps: Examine functional dependencies:

$A_2A_3 \rightarrow A_4$ and $A_1A_5 \rightarrow A_5$

We are looking for if one of them is a superkey.

Closure of $\{A_2, A_3\}$ =

$\{A_2, A_3\}^+ = \{A_2, A_3, A_4\}$ does not include all attributes in R so $\{A_2, A_3\}$ is not a superkey. So not BCNF.

Closure of $\{A_1, A_3\}$ =

$\{A_1, A_3\}^+ = \{A_1, A_3, A_5\}$ does not include all attributes in R, so $\{A_1, A_3\}$ is not a superkey. So not BCNF

Third Normal form (3NF):

$\alpha \rightarrow \beta$ in F^+

at least one of the following holds.

$\alpha \rightarrow \beta$ is trivial ($\beta \in \alpha$)

α is a superkey for R

Each attribute A in $\beta - \alpha$ is contained in a candidate key for R.

If a relation is BCNF it is in 3NF

Because of this relation does not satisfy α is a superkey for $R\{A_2, A_3\}$ and $\{A_2, A_3\}$ it is not 3NF either.

b) I want to use Lossless-join decomposition.

$R_1 \bowtie R_2 = (A_2, A_3, A_4) \bowtie (A_1, A_3, A_5)$

common attribute

$R_1 \cap R_2 = \{A_3\}$

$R_1 \cup R_2 = \{A_1, A_2, A_3, A_4, A_5\}$

I decomposed it that way because I can get all attributes from original relation R.

c) $R_1 = (A_2, A_3, A_4)$

Functional dependencies in R1:

$A_2 A_3 \rightarrow A_4$

Closure of $\{A_2, A_3\}$:

$\{A_2 A_3\}^+ = \{A_2, A_3, A_4\}$ does include all attributes in R1, so $\{A_2, A_3\}$ is a superkey. It's BCNF.

for controlling 3NF:

each attribute A in $\beta - \alpha$ is contained in a candidate key for R. For finding candidate keys.

Steps:

1) Find attributes that are neither on the left and right side.
-none

2) Find attributes that are only on the right side.
-A_4

3) Find attributes that are only on the left side
-A_2, A_3

4) Combine the attributes on step 1 and 3
-Since step 1 has no attributes, it's just A_2, A_3.

5) Test if the closures of attributes on step 4 are all the attributes.

-Yes because with $A_2 A_3$ we get A_4. So it is 3NF.

So we can say that if a relation is BCNF it is also in 3NF

$R_2 = (A_1, A_3, A_5)$

functional dependencies in R_2:

$A_1 A_3 \rightarrow A_5$

Closure of $\{A_1, A_3\}$

$\{A_1, A_3\}+ = \{A_1, A_3, A_5\}$ does include all attributes in R_1. So $\{A_2, A_3\}$ is a superkey. So it is BCNF

For controlling 3NF:

Each attribute A in $\beta - \alpha$ is contained in a candidate key for R. For finding candidate keys,

steps:

1) find the attributes that are neither on left or right side
-none

2) find attributes that are only on the right side.
- A_5

3) find attributes that are only on left side
$-A_1, A_3$

4) Combine attributes on step 1 and 3.
-Since step 1 has no attributes, it's just A_1, A_3

5) Test if closures of attributes on step 4 are all the attributes.
-Yes we can get A_5 with A_1, A_3
So this relation is 3NF.

d) Decomposed condition

$(A_2, A_3, A_4), (A_1, A_3, A_5)$

Dependencies:

$R_1 = A_2, A_3 \rightarrow A_4$ $R_2 = A_1, A_3 \rightarrow A_5$

They are preserved by the decomposition because with $A_2, A_3 \rightarrow A_4$ we get (A_2, A_3, A_4) and with $A_1, A_3 \rightarrow A_5$ we get (A_1, A_3, A_5).

2) NoSQL databases have different types of data models. The main types are document, key-value, wide-column and graph. So I must firstly choose a type for my database system. In a hospital we need to access our datas very fast. In case we have very huge data like patient, doctors, staff infos. So I decided to use columnar database because I have huge data. With hight scalibality in columnar databases database system can handle increasing amounts of data and growing workloads efficiently. And with columnar databases my data can be highly compressed. With this my database uses less disk space. In columnar databases read and write requests handle in parallel which brings high performance. And in columnar databases I want to use Cassandra because it provides all of my wants.

Example entity:

Column Name	Data Type	Description
PatientName	Text	Full name of patient

for inserting single value in patients:

```
INSERT INTO patient (patientNo, name, age, sex, phoneNo, address,
                     date, payment, appointment)
VALUES ('1', 'Alperen Karasete', '25', 'Male', '539 999 99 99', 'Istanbul/Tuzla',
        13-03-2023), '500', '13-05-2023');
```