A) 1NF

We first need to remove the duplicate ‘EmployeeName’ for ‘EmployeeID’ E01.

EmployeeID	EmployeeName
E01	John Doe
E02	Jane Smith
E03	Mike Brown

ProjectID	ProjectName	ProjectDepartment	EmployeeID
P01	Project Alpha	IT	E01
P02	Project Beta	Marketing	E02
P03	Project Gamma	Development	E01
P02	Project Beta	Marketing	E03

2NF & 3NF

Then because that each EmployeeID uniquely identifies ‘EmployeeName’ and each ‘ProjectID’ uniquely identifies ‘ProjectName’ and ‘ProjectDepartment’

Employees

EmployeeID	EmployeeName
E01	“Jane Doe”
E02	“Jane Smith”
E03	“Mike Brown”

Projects

ProjectID	ProjectName	ProjectDepartment
P01	“Project Alpha”	“IT”
P02	“Project Beta”	“Marketing”

P03	“Project Gamma”	“Development”

EmployeeProjects

EmployeeID	ProjectID
E01	P01
E02	P02
E01	P03
E03	P02

B)

To ensure the table adheres to Third Normal Form (3NF), it must first satisfy the conditions of Second Normal Form (2NF) and be devoid of transitive dependencies. This particular table exhibits transitive dependencies, with both the Author Name and Release Date being dependent on the Book Title, which in turn is determined by the Purchase ID. This situation leads to a violation of 3NF. To rectify this and achieve normalization, the table should be divided as follows:

Books

BookID	BookTitle	AuthorName	ReleaseDate
1	“The Great Gatsby”	“F. Scott Fitzgerald”	1925
2	“To Kill a Mockingbird”	“Harper Lee”	1960
3	“1984”	“George Orwell”	1949

Purchases

PurchaseID	BookID
001	1
002	2
003	1

004	3

C)

The table provided fails to meet the requirements of 1NF since the author column contains non-atomic values, comprising multiple author names. Normalization is necessary to address this issue.

Authors

AuthorID	Author
1	“Author X
2	“Author Y”
3	“Author Z”

Books

BookID	BookName	Price	Publisher	Year
1	“Book A”	20	“Publisher Z”	2020
2	“Book B”	15	“Publisher Y”	2022
3	“Book C”	25	“Publisher Z”	2019

Books + Authors

Book ID	AuthorID
1	1
1	2
2	1
2	3
3	2
3	3

2 a)
The relation R(ABCD) with the set of functional dependencies $\{A \rightarrow C, B \rightarrow D, AB \rightarrow CD\}$ is not in Second Normal Form. The presence of the functional dependencies $A \rightarrow C$ and $B \rightarrow D$ indicates partial dependencies, where the non-prime attributes C and D are dependent on parts of the candidate key (A and B, respectively), rather than on the full candidate key AB.

To transform R into 2NF, we eliminate these partial dependencies by decomposing R into three 2NF-compliant relations:

1. $R_1(AC)$, which removes the partial dependency $A \rightarrow C$ by isolating it in a separate relation where A is a prime attribute and C is fully functionally dependent on it.
2. $R_2(BD)$, which similarly isolates the partial dependency $B \rightarrow D$, ensuring D is fully functionally dependent on the prime attribute B.
3. $R_3(AB)$, which retains the original candidate key AB to preserve the relationship between A and B.

This decomposition into R_1, R_2, and R_3 ensures that each relation is in 2NF, with all non-prime attributes being fully functionally dependent on their respective relation's candidate key. Furthermore, the decomposition preserves all original functional dependencies and ensures lossless join property, maintaining the integrity of the original relation R.

2b)

Upon examining the relation R(ABCDE) with functional dependencies $\{A \rightarrow B, AB \rightarrow C, C \rightarrow D, A \rightarrow E\}$, it is determined that A is the sole candidate key since A^+ (the closure of A) includes all other attributes in the relation.

The relation is in 2NF as every non-prime attribute (B, C, D, and E) is fully functionally dependent on the candidate key A. No attribute is dependent on a subset of the candidate key, which satisfies the 2NF condition.

However, when evaluating for 3NF, we observe a violation with the functional dependency $C \rightarrow D$. C is not a superkey, and D is not a prime attribute, hence introducing a transitive dependency. This does not meet the criteria for 3NF, which requires every non-prime attribute to be non-transitively dependent on every candidate key.

To rectify this and achieve 3NF, we decompose R into two relations:

$R_1(CD)$, where C is the candidate key, and D is dependent on C, satisfying the 3NF requirements by removing the transitive dependency.
$R_2(ABCE)$, retaining the original candidate key A and including B and E, which are fully functionally dependent on A, and C, which has become a prime attribute as part of R_1's key.

This decomposition ensures no transitive dependencies exist in either relation, aligning with the 3NF rules and preserving the dependencies of the original relation.

2c)

For relation R(ABCDE) with functional dependencies F = {AB → CDE, CD → ABE, E → D}, we initiate the normalization analysis by determining the candidate keys, which are found to be {AB, CD, CE} after computing closures for all combinations of attributes.

The relation R is in 2NF since all attributes are part of a candidate key; thus, there are no non-prime attributes, eliminating the possibility of partial dependencies where a non-prime attribute is dependent on a subset of a candidate key.

When evaluating for 3NF, R satisfies the criteria as each functional dependency has either a superkey on its left-hand side (AB and CD) or a prime attribute on the right-hand side (E→D, where D is prime). Thus, all functional dependencies conform to the rule that no non-prime attribute should be transitively dependent on a superkey.

However, the relation fails to meet the stringent requirements of BCNF. In BCNF, for any dependency X→Y, X must be a superkey. The dependency E→D does not fulfill this, since E is not a superkey. The FD E→D indicates a dependency where a non-superkey determines a prime attribute, which violates BCNF conditions.

Thus, we establish that the highest normal form R satisfies is 3NF, and it does not ascend to BCNF due to the aforementioned violation.

2d)

To achieve Boyce-Codd Normal Form (BCNF) for relation R(ABCDE), we isolate the functional dependency E→D that violates BCNF, as E is not a superkey. We decompose R into:

R_1(DE) with the functional dependency E→D. E is a superkey in R_1, satisfying BCNF.
R_2(ABCE) retains the rest of the attributes. The dependencies AB→CE and CE→AB in R_2 have superkeys as their antecedents, complying with BCNF.
Post-decomposition, R_1 and R_2 are both in BCNF, with no violations present. This ensures that R is now normalized to BCNF.

3.

a)