Question1

Bank:

Bank table primary key: {Bankname,City} which is combined with two attributes.
Bank table foreign key: Don't have foreign key.

The reason why I use Bankname and City as primary key is because the handout mentions "The banks are specified by the name of the bank and the city where the branch is located in."

Constraint: I set BankName and City not null because they consist a primary key in the table. I used CHECK for make sure Noaccount is a positive number. And I made constraint for security level by using CHECK to make sure the value is inside the 'excellent', 'very good', 'good', 'weak' value.

Robberies:

Robberies table primary key: {BankName,City, Date} three attributes composite as a primary key.

Robberies table foreign key:{BankName, City} $\subseteq$ Banks{BankName,City}

We need to include Date as a primary key ,otherwise we can not distinguish if a bank have been robbed more than once in the same date.

I used DELETE RESTRICT , because I want to avoid happen violate data consistency when delete (BankName,City) in Banks table but other table still referencing (BankName,City).

I used UPDATE CASCADE , When bank change its name , the robberies referencing to the bank should be updated as same time to avoid data consistency.

Constraint: I set BankName and City not null because both of them are used as primary key in robberies table and they are referencing the banks table. And I set Date not null because Date also will be part of primary key. Then I used CHECK for checking the number of money be stolen is positive value.

Plan:

Plans table primary key : {BankName,City,PlannedDate}
PlannedDate can make sure we could find wehter a bank be planed several times.

Plans table foreign key : {BankName,City} $\subseteq$ Banks{BankName,City}

I used DELETE RESTRICT , because I want to avoid happen violate data consistency when delete (BankName,City) in Banks table but other table still referencing (BankName,City)

I used UPDATE CASCADE , When bank change its name , the robberies referencing to the bank should be updated as same time to avoid data consistency.

Constraint: I set not null value for BankName , City and PlannedDate because they are primary key in Plans table.And I made RobberLimit constraint to make sure at least one robber plan to steal in a bank by using CHECK.

Robber:

Robbers table primary key : {RobberId}
Robbers table foreign key : NO foreign key .

robberid is unique and I used Serial to generate it.

Constraint:I created two constraint, first one is ageLimit which is for making sure age will always be positive value. Second one is prisonLimit which will make sure the number of years spent in prison is bigger than 0 and less robber's age.

Skills:

Skills table primary key : {SkillId}
Skills table foreign key : NO foreign key .

SkillId is unique and I used Serial to generate it.

hasSkills:

hasSkills table primary key : {RobberId,SkillId}
In handout, " Each skill of a robber has a preference rank, and a grade".
hasSkills table foreign key: {RobberId} $\subseteq$ Robbers{RobberId}

{SkillId} $\subseteq$ Skills{SkillId}

In this way, robbers table can connect with skills table.The Robberid in hasSkills table is a subset of Robberid in Robbers table, which can make sure hasSkills refer to Robbers.

I used ON DELETE RESTRICT to avoid cause violate data consistency when RobberId or SkillId deleted in either Robbers or Skills table while hasSkills table still holding (RobberId,SkillId) which has already been deleted.

I used ON UPDATE RESTRICT ,it shouldn't be allowed to update the RobberID in the robbers table .

Constraint: I used CHECK for making sure grade is in this set ('A+', 'A', 'A-', 'B+', 'B', 'B-', 'C+', 'C') by creating constraint gradeLimit.And I create preferenceLimit for control the value between 1 to 3.

hasAccounts:

hasAccounts table primary key : {RobberId,BankName,City}
Handout said " which stores information about the banks where individual gang members have accounts. " if we know BankName, City , RobberId attribute , we can get which individual gang member has accounts in particular bank.

hasAccounts table foreign key: {RobberId} ⊆ Robbers{RobberId}

{BankName,City} ⊆ Banks{BankName,City}

These two foreign key will connect hasAccounts table with Banks and Robbers table.

I used ON DELETE RESTRICT for RobberID for avoid cause violate data consistency when RobberId deleted in Robbers table but
hasAccounts table still referencing RobberId which has been already deleted.

I used UPDATE RESTRICT for RobberID .

Constraint: BankName and City and RobberId are important information to this table as a composite primary key so I made them all not null.

Accomlices:

Accomlices table primary key: {RobberId,BankName,City,RobberyDate}
Handout said " stores information about which gang members participated in each bank robbery, and what share of the money they got" After combine, we can get when, where and who.

Accomlices table foregin key:{RobberId} ⊆ Robbers{RobberId}

{BankName,City,RobberyDate} ⊆ Banks{BankName,CityRobberyDate}

An Accomplice should be a Robber who exists in the Robbers table .
An Accomplice data that is inserted should always refer to robberies that exist in Robberies.

I used ON DELETE RESTRICT for RobberID for making sure avoid cause violate data consistency when RobberId deleted in Robbers table but Accomplices table still referencing RobberId which has been already deleted.

I used UPDATE RESTRICT for RobberID.

Delete action applied for {BankName,City,RobberyDate} same reason , but when we updated the those attribute , it should be updated in Banks table first because BankName and City in the Banks table.

Constraint:I set not null value for RobberId , BankName , City , RobberyDate because they are primary keys.And I used constraint to check sharing value of robbers is positive or not.