PART 1:

#1

a) $PQ^+ = \{PQRTSU\}$ ✓ $QR^+ = \{QRTS\}$ ✗ $PR^+ = \{PRU\}$ ✗

$QR \subseteq PQ$

$PR \subseteq PQ$

F_1 is NOT a BCNF

$P^+ = \{PQSTU\}$ ✗ $R^+ = \{RPQSTU\}$ $PQ \subseteq R$ ✓ $PQ^+ = \{PQSTU\}$ ✗ $ST^+ = \{STU\}$ ✗

F_2 is NOT a BCNF

$PQR^+ = \{PQRSTU\}$ ✓ $QR^+ = \{QRTPSU\}$ ✓ $QRT^+ = \{QRTUPS\}$ ✓

$QR \subseteq PQR$

F_3 is a BCNF

b) F_1 : $\langle PQ \rangle$

F_2 : $\langle R \rangle$

F_3 : $\langle QR \rangle$

c) #1

$R_1(P, Q, R, S)$ $R_1 \cap R_2 = \{S\}$ → $R_1 - R_2 = \{P, Q, R\}$

$R_2(S, T, U)$ $R_1 \cap R_2 = \{S\}$ → $R_2 - R_1 = \{T, U\}$

F_1 : $S^+ = \{S\}$ F is lossy

F_2 : $S^+ = \{S\}$ F_2 is lossy

F_3 : $S^+ = \{S\}$ F_3 is lossy

#2

$A_1(P, Q, R)$ $A_1 \cap A_2 = \{\emptyset\}$ → $A_1 - A_2 = \{P, Q, R\}$

$A_2(S, T, U)$ → $A_2 - A_1 = \{S, T, U\}$

This decomposition is lossy for all FD's because the intersection is null which means that there is no common attribute to help maintain the connection between the 2 subrelations.

#3

$A_1(P, Q, R, T)$ $A_1 \cap A_2 = \{Q, R\}$ → $A_1 - A_2 = \{P\ T\}$

$A_2(Q, R, S, U)$ → $A_2 - A_1 = \{S\ U\}$

F_1 : $QR^+ = \{QRTS\}$ ⇒ F_1 is lossy

F_2 : $QR^+ = \{QRPSTU\}$ ⇒ F_2 is lossless

F_3 : $QR^+ = \{QRTP\}$ ⇒ F_3 is lossless

PART 2

Part A:

Relation:
PublicationVenue(Abbr, names, start_year, DBLP_link)
PublicationVenueNames(Abbr, names)
Conference(Abbr, names, start_year, DBLP_link)
Journal(Abbr, names, start_year, DBLP_link)
belongTo(Abbr, field_name, major_name)
researchField(field_name, major_name)
major(major_name)
PublicationRanker(ranker_name)
UniversityRanker(ranker_name)
Ranker(ranker_name)
rankByMajor(major_name, ranker_name, rank, method, year)
rankByField(field_name, ranker_name, major_name, org_name, rank, method, year)
rankPV(ranker_name, Abbr, rank)
organization(org_name, continent, country, state, city, short_name, zipcode)
University(org_name, continent, country, state, city, short_name, zipcode, in_state_student_fee, out_state_student_fee, graduate_enrollment, undergrad_enrollment)
UniversityTypes(org_name, university_types)
UniversityReviews(org_name, reviews)
UniversityPhotos(org_name, photo_id)
industry(org_name, continent, country, state, city, short_name, zipcode)
includes(org_name, major_name)
affiliationUniversity(org_name,ORCID, begin_time, end_time, type)
affiliationIndustry(org_name, ORCID, begin_time, end_time)
author(ORCID, name, google_scholar_link)
publishedBy(ORCID, publication_id)
publicationRecords(publication_id, title)
publishedAt(publication_id, year, Abbr)
publicationEdition(Abbr, year)
journalEdition(Abbr, year)
conferenceEdition(Abbr, year, hotel, end_date, start_date, country, city, age)
generalChairs(ORCID, Abbr, year)
programChairs(ORCID, Abbr, year)

NOT NULL Attributes:
All attributes in PublicationVenue are NOT NULL
All attributes in Conference and Journal table are NOT NULL
All attributes in belongTo are NOT NULL
All attributes in the Research field table are NOT NULL.

All attributes in Organization are NOT NULL
All attributes in University are NOT NULL except for review & photos
All attributes in affiliationUniversity are NOT NULL
All attributes in affiliationIndustry are NOT NULL
Ranker_name is NOT NULL
All attributes in publishedBy are NOT NULL
All attributes in publishedRecords are NOT NULL
All attributes in publishedAt are NOT NULL
All attributes in publishedEdition are NOT NULL
All attributes in journalEdition are NOT NULL
All attributes in conferenceEdition are NOT NULL
All attributes in rankByMajor are NOT NULL
All attributes in rankByField are NOT NULL
All attributes in rankPV are NOT NULL
All attributes in industry are NOT NULL
All attributes in includes are NOT NULL
All attributes in generalChairs are NOT NULL
All attributes in programChairs are NOT NULL

IND:
belongTo[Abbr] ⊆ PublicationVenue[Abbr]
belongTo[field_name,major_name] ⊆ researchField[field_name, major_name]
PublicationVenue[Abbr] ⊆ belongTo[Abbr]
researchField[major_name] ⊆ major[major_name]
rankByMajor[major_name] ⊆ major[major_name]
rankByMajor[ranker_name] ⊆ UniversityRanker[ranker_name]
rankByMajor[org_name] ⊆ University[org_name]
rankByField[field_name, major_name] ⊆ researchField[field_name,major_name]
rankByField[org_name] ⊆ University[org_name]
rankByField[ranker_name] ⊆ UniversityRanker[ranker_name]
rankPV[ranker_name] ⊆ PublicationRanker[ranker_name]
rankPV[Abbr] ⊆ PublicationVenue[Abbr]
PublicationRanker[ranker_name] ⊆ rankPV[ranker_name]
includes[org_name] ⊆ University[org_name]
includes[major_name] ⊆ major[major_name]
University[org_name] ⊆ includes[org_name]
affiliationUniversity[org_name] ⊆ University[org_name]
affiliationUniversity[ORCID] ⊆ author[ORCID]
affiliationIndustry[org_name] ⊆ industry[org_name]
affiliationIndustry[ORCID] ⊆ author[ORCID]
publishedBy[ORCID] ⊆ author[ORCID]
publishedBy[publication_id] ⊆ publicationRecords[publication_id]
publicationRecords[publication_id] ⊆ publishedBy[publication_id]

publishedAt[publication_id] ⊆ publicationRecords[publication_id]
publishedAt[year, Abbr] ⊆ publicationEdition[year,Abbr]
publicationRecords[publication_id] ⊆ publishedAt[publication_id]
publicationEdition[Abbr] ⊆ PublicationVenue[Abbr]
journalEdition[year, Abbr] ⊆ publicationEdition[year,Abbr]
conferenceEdition[year, Abbr] ⊆ publicationEdition[year,Abbr]
generalChairs[ORCID] ⊆ author[ORCID]
generalChairs[year, Abbr] ⊆ conferenceEdition[year, Abbr]
conferenceEdition[year, Abbr] ⊆ generalChairs[year,Abbr]
programChairs[ORCID]⊆ author[ORCID]
programChairs[year, Abbr] ⊆ conferenceEdition[year, Abbr]
conferenceEdition[year, Abbr] ⊆ programChairs[year, Abbr]
PublicationVenueNames[Abbr] ⊆ PublicationVenue[Abbr]
UniversityType[org_name] ⊆ University[org_name]
UniversityReviews[org_name] ⊆ University[org_name]
UniversityPhotos[org_name] ⊆ University[org_name]

Part B:

```
 CREATE TABLE PublicationVenue(
  Abbr VARCHAR(40) NOT NULL,
  start_year INTEGER NOT NULL,
  DBLP_link VARCHAR(40) NOT NULL,
  PRIMARY KEY (Abbr)
);

CREATE TABLE PublicationVenueNames(
  Abbr VARCHAR(40) NOT NULL,
  names VARCHAR(40) NOT NULL,
  PRIMARY KEY (Abbr, names),
  FOREIGN KEY (Abbr) REFERENCES PublicationVenue(Abbr) ON DELETE CASCADE
);

CREATE TABLE Conference(
  Abbr VARCHAR(40) NOT NULL,
  PRIMARY KEY (Abbr),
  FOREIGN KEY (Abbr) REFERENCES PublicationVenue(Abbr) ON DELETE CASCADE
);

CREATE TABLE Journal(
  Abbr VARCHAR(40) NOT NULL,
  PRIMARY KEY (Abbr),
  FOREIGN KEY (Abbr) REFERENCES PublicationVenue(Abbr) ON DELETE CASCADE
);
```

```
CREATE TABLE major(
  major_name VARCHAR(40) NOT NULL,
  PRIMARY KEY (major_name)
);

CREATE TABLE researchField(
  field_name VARCHAR(40) NOT NULL,
  major_name VARCHAR(40) NOT NULL,
  PRIMARY KEY (field_name, major_name),
  FOREIGN KEY (major_name) REFERENCES major(major_name) ON DELETE CASCADE
);

CREATE TABLE belongto(
  Abbr VARCHAR(40) NOT NULL,
  field_name VARCHAR(40) NOT NULL,
  major_name VARCHAR(40) NOT NULL,
  PRIMARY KEY (Abbr),
  FOREIGN KEY (field_name, major_name) REFERENCES researchField(field_name,
major_name) ON DELETE CASCADE
);

CREATE TABLE Ranker(
  ranker_name VARCHAR(40) NOT NULL,
  PRIMARY KEY (ranker_name)
);

CREATE TABLE PublicationRanker(
  ranker_name VARCHAR(40) NOT NULL,
  PRIMARY KEY (ranker_name),
  FOREIGN KEY (ranker_name) REFERENCES Ranker(ranker_name) ON DELETE CASCADE
);

CREATE TABLE UniversityRanker(
  ranker_name VARCHAR(40) NOT NULL,
  year INTEGER NOT NULL,
  method VARCHAR(40),
  PRIMARY KEY (ranker_name,year,method),
  FOREIGN KEY (ranker_name) REFERENCES Ranker(ranker_name) ON DELETE CASCADE
);

CREATE TABLE Organization(
  org_name VARCHAR(40) NOT NULL,
  country VARCHAR(40) NOT NULL,
```

```
  state VARCHAR(40) NOT NULL,
  city VARCHAR(40) NOT NULL,
  short_name VARCHAR(40) NOT NULL,
  zipcode INTEGER NOT NULL,
  PRIMARY KEY (org_name)
);

CREATE TABLE University(
        org_name VARCHAR(40) NOT NULL,
    in_state_student_fee INTEGER NOT NULL,
        out_state_student_fee INTEGER NOT NULL,
        graduate_enrollment INTEGER NOT NULL,
        undergraduate_enrollment INTEGER NOT NULL,
        PRIMARY KEY (org_name),
    FOREIGN KEY (org_name) REFERENCES Organization(org_name) ON DELETE
CASCADE
);

CREATE TABLE industry(
        org_name VARCHAR(40) NOT NULL,
        PRIMARY KEY (org_name),
    FOREIGN KEY (org_name) REFERENCES Organization(org_name) ON DELETE
CASCADE
);

CREATE TABLE includes(
        org_name VARCHAR(40) NOT NULL,
        major_name VARCHAR(40) NOT NULL,
    PRIMARY KEY(org_name, major_name),
        FOREIGN KEY (org_name) REFERENCES University(org_name) ON DELETE
CASCADE,
        FOREIGN KEY (major_name) REFERENCES major(major_name)
);

CREATE TABLE rankByMajor(
  major_name VARCHAR(40) NOT NULL,
  ranker_name VARCHAR(40) NOT NULL,
  org_name VARCHAR(40) NOT NULL,
  `rank` VARCHAR(40) NOT NULL,
  method VARCHAR(40) NOT NULL,
  year INTEGER NOT NULL,
  FOREIGN KEY (major_name) REFERENCES major(major_name),
  FOREIGN KEY (ranker_name, year, method) REFERENCES UniversityRanker(ranker_name,
year, method),
```

```
  FOREIGN KEY (org_name) REFERENCES University(org_name) ON DELETE CASCADE
);

CREATE TABLE rankByField(
  field_name VARCHAR(40) NOT NULL,
  major_name VARCHAR(40) NOT NULL,
  ranker_name VARCHAR(40) NOT NULL,
  org_name VARCHAR(40) NOT NULL,
  `rank` VARCHAR(40) NOT NULL,
  method VARCHAR(40) NOT NULL,
  year INTEGER NOT NULL,
  FOREIGN KEY (field_name, major_name) REFERENCES researchField(field_name,
major_name),
  FOREIGN KEY (ranker_name, year, method) REFERENCES UniversityRanker(ranker_name,
year, method),
  FOREIGN KEY (org_name) REFERENCES University(org_name) ON DELETE CASCADE
);

CREATE TABLE rankPV(
        ranker_name VARCHAR(40) NOT NULL,
        Abbr VARCHAR(40) NOT NULL,
        `rank` VARCHAR(40) NOT NULL,
   PRIMARY KEY(Abbr, ranker_name),
        FOREIGN KEY (ranker_name) REFERENCES PublicationRanker(ranker_name) ON
DELETE CASCADE,
        FOREIGN KEY (Abbr) REFERENCES PublicationVenue(Abbr) ON DELETE CASCADE
);

CREATE TABLE UniversityTypes(
        org_name VARCHAR(40) NOT NULL,
        university_types VARCHAR(40) NOT NULL,
   PRIMARY KEY(org_name, university_types),
        FOREIGN KEY (org_name) REFERENCES University(org_name) ON DELETE
CASCADE
);

CREATE TABLE UniversityReviews(
        org_name VARCHAR(40) NOT NULL,
        reviews VARCHAR(40),
   PRIMARY KEY (org_name, reviews),
        FOREIGN KEY (org_name) REFERENCES University(org_name) ON DELETE
CASCADE
);
```

```
CREATE TABLE UniversityPhotos(
  org_name VARCHAR(40) NOT NULL,
  photo_id INTEGER NOT NULL,
  photos BLOB NOT NULL,
  PRIMARY KEY (org_name, photo_id),
  FOREIGN KEY (org_name) REFERENCES University(org_name) ON DELETE CASCADE
);


CREATE TABLE author(
    orcid INTEGER NOT NULL,
    name VARCHAR(40) NOT NULL,
    google_scholar_link VARCHAR(40) NOT NULL,
    PRIMARY KEY (orcid)
);

CREATE TABLE affiliationUniversity(
        org_name VARCHAR(40) NOT NULL,
        ORCID INTEGER NOT NULL,
        begin_time VARCHAR(40) NOT NULL,
        end_time VARCHAR(40) NOT NULL,
        type VARCHAR(40) NOT NULL,
    PRIMARY KEY(org_name, ORCID),
        FOREIGN KEY (org_name) REFERENCES University(org_name) ON DELETE
CASCADE,
        FOREIGN KEY (ORCID) REFERENCES author(ORCID) ON DELETE CASCADE
);

CREATE TABLE affiliationIndustry(
        org_name VARCHAR(40) NOT NULL,
        ORCID INTEGER NOT NULL,
        begin_time VARCHAR(40) NOT NULL,
        end_time VARCHAR(40) NOT NULL,
        type VARCHAR(40) NOT NULL,
    PRIMARY KEY(org_name, ORCID),
        FOREIGN KEY (org_name) REFERENCES industry(org_name) ON DELETE
CASCADE,
        FOREIGN KEY (ORCID) REFERENCES author(ORCID) ON DELETE CASCADE
);

CREATE TABLE publicationRecords(
    publication_id INTEGER NOT NULL,
    title VARCHAR(40) NOT NULL,
```

```
    PRIMARY KEY (publication_id)
);

CREATE TABLE publishedBy(
    orcid INTEGER NOT NULL,
    publication_id INTEGER NOT NULL,
    PRIMARY KEY(orcid, publication_id),
    FOREIGN KEY(orcid) REFERENCES author(orcid),
    FOREIGN KEY(publication_id) REFERENCES publicationRecords(publication_id)
);

CREATE TABLE publicationEdition(
    Abbr VARCHAR(40) NOT NULL,
    years INTEGER NOT NULL,
    PRIMARY KEY(Abbr, years),
    FOREIGN KEY(Abbr) REFERENCES PublicationVenue(Abbr) ON DELETE CASCADE
);

CREATE TABLE journalEdition(
        Abbr VARCHAR(40) NOT NULL,
    years INTEGER NOT NULL,
    PRIMARY KEY(Abbr, years),
    FOREIGN KEY(Abbr, years) REFERENCES publicationEdition(Abbr,years) ON DELETE
CASCADE
);

CREATE TABLE conferenceEdition(
        Abbr VARCHAR(40) NOT NULL,
    years INTEGER NOT NULL,
    hotel VARCHAR(50) NOT NULL,
    end_date DATE NOT NULL,
    start_date DATE NOT NULL,
    country VARCHAR(50) NOT NULL,
    city VARCHAR(40) NOT NULL,
    age INTEGER NOT NULL,
    PRIMARY KEY(Abbr, years),
    FOREIGN KEY(Abbr, years) REFERENCES publicationEdition(Abbr, years) ON DELETE
CASCADE
);

CREATE TABLE publishedAt(
    publication_id INTEGER NOT NULL,
    years INTEGER NOT NULL,
    Abbr VARCHAR(40) NOT NULL,
```

```
    PRIMARY KEY(publication_id),
    FOREIGN KEY(publication_id) REFERENCES publicationRecords(publication_id),
    FOREIGN KEY(Abbr, years) REFERENCES publicationEdition(Abbr, years)
);

CREATE TABLE generalChairs(
    orcid INTEGER NOT NULL,
    Abbr VARCHAR(40) NOT NULL,
    years INTEGER NOT NULL,
    PRIMARY KEY(orcid, Abbr, years),
    FOREIGN KEY(orcid) REFERENCES author(orcid),
    FOREIGN KEY(Abbr, years) REFERENCES conferenceEdition(Abbr, years) ON DELETE
CASCADE
);

CREATE TABLE programChairs(
    orcid INTEGER NOT NULL,
    Abbr VARCHAR(40) NOT NULL,
    years INTEGER NOT NULL,
    PRIMARY KEY(orcid, Abbr, years),
    FOREIGN KEY(orcid) REFERENCES author(orcid),
    FOREIGN KEY(Abbr, years) REFERENCES conferenceEdition(Abbr, years) ON DELETE
CASCADE
);
```