#3 Relational Model

9 Relationships, 7 entities 2 ISA

Applicant(name: char(30), email: char(30), phone#: integer, address: char(30))
- Candidate Keys: phone#

create_account(**email**: char(30), acc#**:** integer)
- Acc# must be unique
- Candidate Keys:

AppliesFor(**applyemail**: char(30), **applyreferenceID**: Integer)

AcceptDenyOffer(acceptOLEmployee#: Integer, acceptStartDate: Integer, **acceptEmail:** char(30))
- Candidate Keys:

Produce(produceApp#: Integer, ApplyDate: Integer, **produceEmail:** char(30))
- Candidate Keys:

Draft(**draftOLEmployee#**: Integer, **draftEmployee#**: Integer)
- Candidate Keys:

JobListing(positionName: char(30), referenceID: Integer, #ofSpots: Integer, duties: char(2000), salary: Integer, Qualifications: char(2000), ShiftSchedule: Char(300))
- Candidate Keys:

Creates(**referenceID**: Integer, **employee#**: Integer)
- Candidate Keys:

Employer(empphone#: Integer, empName: Char(30), empEmail: Char(30), employee#: Integer, primaryPosition: char(30), team: char(30))
- Candidate Keys: empEmail, empphone#

Conducts(**employee#**: Integer, **interviewer**: Char(30), **interviewee**: Char(30), **Date**: Integer)
- Candidate Keys:

Interview(interviewer: Char(30), interviewee: Char(30), Date: Integer)
- Candidate Keys:

FieldSupervisor(fieldProject: Char(30), **employee#**: Integer)
- Candidate Keys:

HiringManager(department: Char(30), <u>**employee#**</u>: Integer)
- Candidate Keys:

StoresApplication(<u>App#</u>: Integer**, storeAcc#**: Integer, applyDate: Integer)
- Candidate Keys:

CoverLetter(<u>**App#**</u>: Integer, introduction: Char(300))
- Candidate Keys:

Resume(<u>**App#**</u>: integer, education: Char(100), resName: Char(30), experience: Char(500))
- Candidate Keys:

Reviews(<u>**revApp#**</u>: Integer, <u>**revEmployee#**</u>: Integer)
- Candidate Keys:

#4 & 5 Functional Dependencies & Normalization

Employer(empphone#: Integer, empName: Char(30), empEmail: Char(30), <u>employee#</u>: Integer, primaryPosition: char(30), team: char(30))

Employer:

Employee# -> EmpName, PrimaryPosition, Team

PrimaryPosition, EmpName -> Team

EmpEmail -> EmpName, EmpPhone#

Team, EmpEmail -> PrimaryPosition

<u>Simplify FD's:</u>

Employee# -> EmpName

Employee# -> PrimaryPosition

// removed Employee# -> Team

- *(Employee# -> PrimaryPosition, Employee# -> EmpName,* PrimaryPosition, EmpName -> Team)

EmpEmail -> EmpName

EmpEmail -> EmpPhone#

EmpEmail -> PrimaryPosition
// removed Team from LHS

- *Redundancy: (EmpEmail -> EmpName, EmpEmail -> PrimaryPosition,* PrimaryPosition, EmpName -> Team)

PrimaryPosition, EmpName -> Team

<u>Normalization:</u>

Employee#+ = {Employee#, EmpName, PrimaryPosition, Team}
EmpEmail+ = {EmpEmail, EmpName, EmpPhone#}
(Team, EmpEmail)+ = {PrimaryPosition, Team, EmpEmail, EmpPhone#, EmpName}
(PrimaryPosition, EmpName)+ = {PrimaryPosition, EmpName}

Key: (Employee#, EmpEmail)+ = {Employee#, PrimaryPosition, Team, EmpEmail, EmpPhone#, EmpName}

R(empphone#: Integer, empName: Char(30), empEmail: Char(30), <u>employee#</u>: Integer, primaryPosition: char(30), team: char(30))

R1(PrimaryPosition, EmpName, Team) R2(PrimaryPosition, EmpName, Employee#, EmpEmail, EmpPhone#)
R3(Employee#, EmpName) R4(PrimaryPosition, Employee#, EmpEmail, EmpPhone#)
R5(Employee#, PrimaryPosition) R6(Employee#, EmpEmail, EmpPhone#)
R7(EmpEmail, EmpPhone#)R8(Employee#, EmpEmail)

Solution: R1(<u>PrimaryPosition</u>, <u>EmpName</u>, Team), **R3**(<u>Employee#</u>, EmpName)
R5(<u>Employee#</u>, PrimaryPosition), **R7**(<u>EmpEmail</u>, EmpPhone#) **R8**(<u>Employee#</u>, <u>EmpEmail</u>)

Job Listing:

Duties -> PositionName

Qualifications -> Duties

ReferenceID -> PositionName, Duties, Qualifications, #ofSpots

ShiftSchedule -> Salary

<u>Simplify FD's:</u>

// removed ReferenceID -> PositionName
- *Redundancy: (ReferenceID -> Qualifications -> Duties -> PositionName)*

// removed ReferenceID -> Duties
- *Redundancy: (ReferenceID -> Qualifications -> Duties)*

ShiftSchedule -> Salary

ReferenceID -> #ofSpots

Duties -> PositionName

Qualifications -> Duties

ReferenceID -> Qualifications

Normalization

ReferenceID+ = {ReferenceID, Qualifications, #ofSpots, Duties, PositionName}
ShiftSchedule+ = {ShiftSchedule, Salary}
Duties+ = {Duties, PositionName}
Qualifications+ = {Qualifications, Duties, PositionName}

Key: (ReferenceID, ShiftSchedule)+ = {ReferenceID, Qualifications, #ofSpots, Duties, PositionName, ShiftSchedule, Salary}

R(positionName: char(30), referenceID: Integer, #ofSpots: Integer, duties: char(2000), salary: Integer, Qualifications: char(2000), ShiftSchedule: Char(300))

R1(ShiftSchedule, Salary) R2(ShiftSchedule, ReferenceID, Qualifications, #ofSpots, Duties, PositionName)
R3(ReferenceID, #ofSpots) R4(ReferenceID, ShiftSchedule, Qualifications, Duties, PositionName)
R5(Duties, PositionName) R6(ReferenceID, ShiftSchedule, Qualifications, Duties)
R7(Qualifications, Duties) R8(ReferenceID, ShiftSchedule, Qualifications)
R9(ReferenceID, Qualifications), R10(ReferenceID, ShiftSchedule)

Solution:
R1(ShiftSchedule, Salary) **R3**(ReferenceID, #ofSpots)
R5(Duties, PositionName), **R7**(Qualifications, Duties), **R9**(ReferenceID, Qualifications)**,**
R10(ReferenceID, ShiftSchedule)

—---

#6 SQL DDL

Applicant(name: char(30), email: char(30), phone#: integer, address: char(30))

- Candidate Keys: phone#

```
# table for entity Applicant

CREATE TABLE Applicant (
        applicant-email   CHAR(30) PRIMARY KEY,
        name               CHAR(30),
        phone#             INTEGER UNIQUE,
        address            CHAR(30)
);
```

create_account(name: char(30), **email**: char(30), phone#: integer, address: char(30), **acc#:** integer)

- Acc# must be unique
- Candidate Keys: phone#

```
# table combining Creates and Account

CREATE TABLE CreateAccount(
        applicant-email    CHAR(30),
        account-acc#       INTEGER  PRIMARY KEY,
        FOREIGN KEY (applicant-email) REFERENCES Applicant(Email)
);
```

StoresApplication(App#: Integer**, storeAcc#**: Integer, applyDate: Integer)

- Candidate Keys:

```
# table combining Job Application and Stores as one-to-many

CREATE TABLE StoreApplication(
        job-app#         INTEGER PRIMARY KEY
        ApplyDate        INTEGER
        account-acc#     INTEGER
        FOREIGN KEY (account-acc#) REFERENCES CreateAccount(Acc#)
);
```

Produce(produceApp#: Integer, ApplyDate: Integer, **produceEmail:** char(30))

- Candidate Keys:

```
CREATE TABLE ProduceApplication(
        produceApp#         INTEGER PRIMARY KEY,
        ApplyDate           INTEGER,
        produceEmail        CHAR(30),
        FOREIGN KEY(produceEmail) REFERENCES Applicant(Email)
)
```

CoverLetter(**<u>App#</u>**: Integer, introduction: Char(300))

- Candidate Keys:

```
# table for ISA subclass Cover Letter

CREATE TABLE CoverLetter(
        job-app#        CHAR(30) PRIMARY KEY
        introduction    CHAR(300)
        FOREIGN KEY(job-app#) REFERENCES StoreApplication(App#)
);
```

Resume(**<u>App#</u>**: integer, education: Char(100), resName: Char(30), experience: Char(500))

- Candidate Keys:

```
# table for ISA subclass Resume

CREATE TABLE Resume(
        job-app#        CHAR(30) PRIMARY KEY
        education       CHAR(300)
        experience      CHAR(300)
        resName         CHAR(30)
        FOREIGN KEY(job-app#) REFERENCES StoreApplication(App#)
);
```

AppliesFor(**<u>applyEmail</u>**: char(30), **<u>applyReferenceID</u>**: Integer)

```
CREATE TABLE AppliesFor(
        applyEmail              CHAR(30),
        applyReferenceID        INTEGER,
        PRIMARY KEY(applyEmail, applyReferenceID)
        FOREIGN KEY(applyEmail) REFERENCES Applicant(applyEmail),
        FOREIGN KEY(applyReferenceID) REFERENCES JobListing(ReferenceID),
)
```

AcceptDenyOffer(<u>acceptOLEmployee#</u>: Integer, acceptStartDate: Integer, **acceptEmail: char(30)**)

```
#table combining relationship Accepts/Denies with entity Offer Letter
CREATE TABLE AcceptDenyOffer(
        offer-employee#    INTEGER PRIMARY KEY,
        StartDate          INTEGER,
        applicant-email    CHAR(30),
        FOREIGN KEY(applicant-email) REFERENCES Applicant(email)
);
```

Draft(**draftOLEmployee#**: Integer, **draftEmployee#**: Integer)

- Candidate Keys:

```
# relationship table of Draft for Offer Letter+Employer
CREATE TABLE Draft(
        offer-employee#  INTEGER,
        emp-employee#  INTEGER,
        PRIMARY KEY(offer-employee#, emp-employee#)
        FOREIGN KEY(offer-employee#) REFERENCES AcceptDenyOffer(OLEmployee#)
        FOREIGN KEY(emp-employee#) REFERENCES Employer(Employee#)
);
```

Creates(**referenceID**: Integer, **employee#**: Integer)

- Candidate Keys:

```
# relationship table of Creates for Job Listing+Employer
CREATE TABLE Creates(
        job-referID            INTEGER,
        emp-employee#          INTEGER,
        PRIMARY KEY(job-referID, emp-employee#)
        FOREIGN KEY (job-referID) REFERENCES JobListing(referenceID)
        FOREIGN KEY (emp-employee#) REFERENCES Employer(Employee#)
);
```

Employer

R1(PrimaryPosition: Char(30), EmpName: Char(30), Team: Char(30))
R3(Employee#: Integer, EmpName: Char(30))
R5(Employee#: Integer, PrimaryPosition: Char(30))
R7(EmpEmail: Char(30), EmpPhone#: Integer)
R8(Employee#: Integer, EmpEmail: Char(30))

```
CREATE TABLE R1(
PrimaryPosition CHAR(30),
EmpName CHAR(30),
Team CHAR(30),
PRIMARY KEY(PrimaryPosition, EmpName)
)

CREATE TABLE R3(
Employee# INTEGER PRIMARY KEY,
EmpName CHAR(30)
)

CREATE TABLE R5(
Employee# INTEGER PRIMARY KEY,
PrimaryPosition CHAR(30)
)

CREATE TABLE R7(
EmpEmail CHAR(30) PRIMARY KEY,
EmpPhone#: INTEGER,
)

CREATE TABLE R8(
Employee# INTEGER,
EmpEmail CHAR(30),
PRIMARY KEY(Employee#, EmpEmail)
)
```

FieldSupervisor(fieldProject: Char(30), **employee#**: Integer)

- Candidate Keys:

```
# tabel for ISA subclass Supervisor
CREATE TABLE Supervisor(
        emp-employee#    INTEGER  PRIMARY KEY,
        fieldProject     CHAR(30),
        FOREIGN KEY (emp-employee#) REFERENCES Employer(Employee#)
);
```

HiringManager(department: Char(30), **employee#**: Integer)

- Candidate Keys:

```
# tabel for ISA subclass HiringManager
CREATE TABLE HiringManager(
        emp-employee#    INTEGER  PRIMARY KEY,
        department       CHAR(30),
          FOREIGN KEY (emp-employee#) REFERENCES Employer(Employee#)
);
```

Conducts(**employee#**: Integer, **interviewer**: Char(30), **interviewee**: Char(30), **Date**: Integer)

- Candidate Keys:

# relationship table of relationship Conducts for Employer+Interview

```
CREATE TABLE Conducts(
          emp-employee#  INTEGER,
          date                   INTEGER,
          interviewer         CHAR(30),
          Interviewee        CHAR(30),
          PRIMARY (emp-employee#, date, interviewer, Interviewee)
          FOREIGN KEY (emp-employee#) REFERENCES Employer(Employee#)
          FOREIGN KEY (date) REFERENCES Interview(date)
          FOREIGN KEY (interviewer) REFERENCES Interview(interviewer)
          FOREIGN KEY (Interviewee) REFERENCES Interview(Interviewee)
);
```

Interview(interviewer: Char(30), interviewee: Char(30, date: Integer)

- Candidate Keys:

```
CREATE TABLE Interview(
          date                   INTEGER,
          interviewer         CHAR(30),
          Interviewee        CHAR(30),
          PRIMARY KEY (date, interviewer, Interviewee)
);
```

Review(**revApp#**: Integer, **revEmployee#**: Integer)

- Candidate Keys:

# table for relationship Reviews for Job Application+Employer

```
CREATE TABLE Reviews(
          job-app#               INTEGER,
          emp-employee#    INTEGER,
          PRIMARY KEY (job-app#, emp-employee#)
          FOREIGN KEY (job-app# ) REFERENCES StoreApplication(App#)
          FOREIGN KEY (emp-employee#) REFERENCES Employer(Employee#)
)
```

JobListing

R1(ShiftSchedule: Char(30), Salary: Integer)

R3(ReferenceID: Integer, #ofSpots: Integer)

R5(Duties: Char(3000), PositionName: Char(30)),
R7(Qualifications: Char(3000), Duties: Char(3000)),
R9(ReferenceID: Integer, Qualifications: Char(3000))**,**
R10(ReferenceID: Integer, ShiftSchedule: Char(3000))

```
CREATE TABLE R1(
ShiftSchedule CHAR(30) PRIMARY KEY,
Salary INTEGER
)

CREATE TABLE R3(
ReferenceID INTEGER PRIMARY KEY,
#ofSpots INTEGER
)

CREATE TABLE R5(
Duties CHAR(3000) PRIMARY KEY,
PositionName CHAR(30)
)

CREATE TABLE R7(
Qualifications CHAR(3000) PRIMARY KEY,
Duties CHAR(3000)
)

CREATE TABLE R9(
ReferenceID INTEGER PRIMARY KEY,
Qualifications CHAR(3000)
)

CREATE TABLE R10(
ReferenceID INTEGER,
ShiftSchedule CHAR(30),
PRIMARY KEY(ReferenceID, ShiftSchedule)
)
```

LIST OF TABLES AFTER NORMALIZATION

Applicant(name: char(30), email: char(30), phone#: integer, address: char(30))
- Candidate Keys: phone#

create_account(**email**: char(30), acc#**:** integer)
- Acc# must be unique

- Candidate Keys:

AppliesFor(**<u>applyemail</u>**: char(30), **<u>applyreferenceID</u>**: Integer)

AcceptDenyOffer(<u>acceptOLEmployee#</u>: Integer, acceptStartDate: Integer, **acceptEmail:** char(30))

- Candidate Keys:

Produce(<u>produceApp#</u>: Integer, ApplyDate: Integer, **produceEmail:** char(30))

- Candidate Keys:

Draft(**<u>draftOLEmployee#</u>**: Integer, **<u>draftEmployee#</u>**: Integer)

- Candidate Keys:

JobListing:

R1(<u>ShiftSchedule</u>: Char(30), Salary: Integer)
R3(<u>ReferenceID</u>: Integer, #ofSpots: Integer)
R5(<u>Duties</u>: Char(3000), PositionName: Char(30)),
R7(<u>Qualifications</u>: Char(3000), Duties: Char(3000)),
R9(<u>ReferenceID</u>: Integer, Qualifications: Char(3000))**,**
R10(<u>ReferenceID</u>: Integer, <u>ShiftSchedule</u>: Char(3000))

Creates(**<u>referenceID</u>**: Integer, **<u>employee#</u>**: Integer)

- Candidate Keys:

Employer:

R1(<u>PrimaryPosition</u>: Char(30), <u>EmpName</u>: Char(30), Team: Char(30))
R3(<u>Employee#:</u> Integer, EmpName: Char(30))
R5(<u>Employee#:</u> Integer, PrimaryPosition: Char(30))
R7(<u>EmpEmail:</u> Char(30), EmpPhone#: Integer)
R8(<u>Employee#</u>: Integer, <u>EmpEmail</u>: Char(30))

- Candidate Keys: empphone#

Conducts(**<u>employee#</u>**: Integer, **<u>interviewer</u>**: Char(30), **<u>interviewee</u>**: Char(30), **<u>Date</u>**: Integer)

- Candidate Keys:

Interview(<u>interviewer</u>: Char(30), <u>interviewee</u>: Char(30, <u>date</u>: Integer)

- Candidate Keys:

FieldSupervisor(fieldProject: Char(30), **<u>employee#</u>**: Integer)

- Candidate Keys:

HiringManager(department: Char(30), **<u>employee#</u>**: Integer)

- Candidate Keys:

StoresApplication(App#: Integer, **storeAcc#**: Integer, applyDate: Integer)
- Candidate Keys:

CoverLetter(**App#**: Integer, introduction: Char(300))
- Candidate Keys:

Resume(**App#**: integer, education: Char(100), resName: Char(30), experience: Char(500))
- Candidate Keys:

Reviews(**revApp#**: Integer, **revEmployee#**: Integer)
- Candidate Keys: