Alan Ou
301320642

Part 1

```
CREATE TABLE Rep (
        rsin            CHAR(9),
        fname           CHAR(20),
        lname           CHAR(20) NOT NULL,
        studio          CHAR(20),
        phone           CHAR(10),
        PRIMARY KEY (rsin)
)

CREATE TABLE plays (
        msin            CHAR(9),
        aname           CHAR(20),
        share           REAL,
        rsin            CHAR(9),
        PRIMARY KEY (msin, aname),
        FOREIGN KEY (rsin) REFERENCES Rep,
        FOREIGN KEY (aname) REFERENCES Artist,
        FOREIGN KEY (msin) REFERENCES Musician,
        CONSTRAINT unique_email UNIQUE (email)
)

CREATE TABLE Artist (
        aname           CHAR(20),
        startdate       DATETIME,
        genre           CHAR(20),
```

```
        PRIMARY KEY (aname),
)

CREATE TABLE Musician (
        msin        CHAR(9),
        fname       CHAR(20),
        lname       CHAR(20) NOT NULL,
        instrument  CHAR(20),
        email       CHAR(10),
        PRIMARY KEY (msin),
        CONSTRAINT unique_email UNIQUE (email)
)

CREATE TABLE Song (
        isrc        CHAR(12),
        title       CHAR(20),
        year        INTEGER,
        duration    REAL,
        album       CHAR(20),
        aname       CHAR(20) NOT NULL,
        PRIMARY KEY (isrc),
        FOREIGN KEY (aName) REFERENCES Artist
)

CREATE TABLE Sales (
        month       CHAR(9),
        year        INTEGER,
```

```
    vendor      CHAR(20),
    country     CHAR(20),
    amount      REAL,
    quantity    INTEGER,
    isrc        CHAR(12),
    PRIMARY KEY (month, year, vendor, country, isrc),
    FOREIGN KEY (isrc) REFERENCES Song ON DELETE CASCADE
)

CREATE TABLE writes (
    wsin        CHAR(9),
    royalty     REAL,
    isrc        CHAR(12),
    PRIMARY KEY (wsin, isrc),
    FOREIGN KEY (isrc) REFERENCES Song,
    FOREIGN KEY (wsin) REFERENCES Writer
)

CREATE TABLE Writer (
    wsin        CHAR(9),
    fname       CHAR(20),
    lname       CHAR(20) NOT NULL,
    PRIMARY KEY (wsin),
)
```

Part 2

a) $\pi_{\text{firstName, lastName}} (\sigma_{\text{birthDate} < 1985\text{-}01\text{-}01 \wedge \text{income} > 75000} (\text{Customer}))$

Alan Ou
301320642

b) $\pi_{customerID, lastName, income}$ ($\sigma_{customer.customerID = owns.customerID \wedge owns.accNumber = account.accNumber \wedge account.branchNumber = branch.branchNumber \wedge budget > 1000000}$ (Customer x Owns x Account x Branch))

c) $\pi_{employee.sin, firstName, lastName, salary}$ ($\sigma_{employee.sin = branch.managerSIN \wedge employee.sin = personalBanker.sin}$ (Employee x Branch x PersonalBanker))

d) $\pi_{owns.customerID, owns.accNumber}$ (Owns $\bowtie_{owns.customerID = d.customerID \wedge owns.accNumber = d.accNumber}$ ρ_d (Owns))

e) $\pi_{sin, salary}$ ($\sigma_{d.sin = managerSIN \wedge d.branchNumber = employee.branchNumber \wedge employee.salary > d.salary}$ (Employee x ρ_d (Employee) x branch))

f) $\pi_{branchName}$ (Employee $\bowtie_{employee.branchNumber = branch.branchNumber \wedge employee.lastName = \text{“Taylor”} \wedge employee.lastName = \text{“Smith”}}$ Branch)

g) $\pi_{firstName, lastName, birthDate}$ ($\sigma_{customer.customerID = owns.customerID \wedge owns.accNumber = account.accNumber \wedge account.branchNumber = branch.branchNumber \wedge branchName = \text{“Metrotown”}}$ (Customer x Account x Owns x Branch)) $\cup$ $\pi_{firstName, lastName, startDate}$ (Employee $\bowtie_{employee.branchNumber = branch.branchNumber \wedge branchName = \text{“Metrotown”}}$ Branch)

h) $\pi_{customer.customerID, birthDate}$ ($\sigma_{customer.customerID = personalBanker.customerID \wedge personalBanker.sin = employee.sin \wedge employee.branchNumber = branch.branchNumber \wedge branch.branchName = \text{“Lonsdale”}}$ (Customer x Employee x Branch) $\cap$ $\pi_{customer.customerID, birthDate}$ ($\sigma_{customer.customerID = personalBanker.customerID \wedge}$

personalBanker.sin = employee.sin ∧ employee.branchNumber = branch.branchNumber ∧ branch.branchName = "Broadway" (Customer x Employee x Branch)

i) $\pi_{customer.customerID}$ (Customer ⋈ Transaction ⋈ Owns) – $\pi_{customer.customerID}$ ($\sigma_{transaction.amount \geq -10000 \vee Transaction.amount \leq 10000}$ (Customer ⋈ Transaction ⋈ Owns))

j) $\pi_{customer.customerID,\ income}$ (Customer ⋈ Account ⋈ Owns ÷ π_{type} (Account))

k) $\pi_{sin,\ firstName,\ lastName}$ (Customer ⋈ Employee ⋈ Owns ⋈ Account ⋈ Branch)

Although this query may produce the desired data some of the time, there are times where if the employee's first and last name match another customer's first and last name in the database, then the employee may take that customer's customerID, instead of their own, producing wrongly matched data, which would be undesirable.

Part 3

a) {t | ∃c∈Customer (c.income > 75000 ∧ c.birthDate < 1985-01-01 ∧ t.firstName = c.firstName ∧ t.lastName = c.lastName)}

c) {t | ∃e∈Employee ∃p∈PersonalBanker ∃b∈Branch (p.sin = b.managerSIN ∧ t.sin = e.sin ∧ t.firstName = e.firstName ∧ t.lastName = e.lastName ∧ t.salary = e.salary)}