NYU Tandon School of Engineering **Oct 5, 2017**
Computer Science and Engineering
CS6083, Fall 2017

Problem Set #1 Sample Solution

Problem1·

(a) AritistGenre.aid references Aritst.aid
Concert.vid references Venue.vid
ConcertArtist.cid references Concert.cid
ConcertArtist.aid references Artist.aid
TicketType.cid references Concert.cid

(b) The (vid, cdate, cstarttime) would be a suitable PK.

```
(c)  I.  SELECT aname FROM Artist NATURAL JOIN ArtistGenre WHERE genre = “Jazz”

    II.  SELECT DISTINCT vcity
         FROM Venue NATURAL JOIN Concert NATURAL JOIN
         ConcertArtist NATURAL JOIN Artist
         WHERE aname = “Bruno Mars” AND YEAR(cdate) = 2016

   III.  SELECT vid, vname, Count(*)
         FROM Venue V LEFT OUTER JOIN Concert C ON V.vid = C.vid
         WHERE YEAR(C.cdate) = 2016
         Group by vid,vname

    IV.  SELECT cid, ticketclass, ticketcost, cdate
         FROM TicketType NATURAL JOIN Concert NATURAL JOIN ConcertArtist
                   NATURAL JOIN ArtistGenre
         WHERE genre = “Jazz” and
                                 ticketcost = (SELECT MAX(ticketcost)
                                 FROM TicketType NATURAL JOIN Concert NATURAL JOIN
                                 ConcertArtist NATURAL JOIN ArtistGenre
                                 WHERE genre = “Jazz”)
    V. SELECT aid, aname
       FROM Artist NARTURAL JOIN ConcertArtist
       WHERE ananme <> “Bruno Mars” AND cid in
       (SELECT cid FROM Concert NATURAL JOIN ConcertArtist NATURAL JOIN Artist
        WHERE name = “Bruno Mars” and YEAR(cdate) = 2016)
       GROUP BY aid, aname
       HAVING COUNT(*) >= 2
```

```
VI. SELECT aid, aname
      FROM Artist
      WHERE aid NOT IN
      (SELECT aid
       FROM Concert NATURAL JOIN Venue NATURAL JOIN ConsertArtist
       WHERE vcity = “New York City”)

VII. SELECT cid, ticketcost, cdate
      FROM TicketType NATURAL JOIN Concert NATURAL JOIN Venue NATURAL JOIN
              ConcertArtist NATURAL JOIN ArtistGenre
      WHERE  genre = “Jazz” AND vcity = “Chicago” AND YEAR(cdate) = 2017
               AND MONTH(cdate) =  “October” AND
               ticketcost = (SELECT MIN(ticketcost)
                             FROM TicketType NATURAL JOIN Concert NATURAL JOIN Venue
                                    NATURAL JOIN ConcertArtist NATURAL JOIN ArtistGenre
                             WHERE genre = “Jazz” AND vcity = “Chicago” AND
                                    YEAR(cdate) = 2017 AND MONTH(cdate) = “October”)
```

(d) Answer:

I. $\Pi_{aname}\sigma_{genre = "Jazz"}(Artist \bowtie ArtistGenre)$

II. $\Pi_{vcity}\sigma_{aname = "Bruno\ Mars" \wedge Year(cdate) = 2016}(\ Venue \bowtie Concert\ \ \bowtie ConcertArtist \bowtie Artist)$

III. ${}_{vid,\ vname}G_{count(*)}(\sigma_{\ Year(cdate) = 2016}\ (Venue \mathbin{⟗} Concert))$

IV. temp ←

$G_{MAX(ticketcost)\ as\ maxPrice}(\sigma_{genre = "Jazz"}\ (TicketType \bowtie Concert \bowtie ConcertArtist \bowtie ArtistGenre))$

$\Pi_{cid,\ ticketclass,\ ticketcost,\ cdate}\sigma_{\ genre = "Jazz"\ \ \wedge ticketcost = maxPrice}(($
$TicketType \bowtie Concert \bowtie ConcertArtist \bowtie ArtistGenre)\times temp)$

V. temp ← $\Pi_{cid}\sigma_{aname = "Bruno\ Mars" \wedge Year(cdate) = 2016}\ (ConcertArtist \bowtie Artist \bowtie Concert)$

temp2 ← ${}_{aid,\ aname}G_{count(*)\ as\ ccount}(\sigma_{aname <> "Bruno\ Mars"}\ (Artist \bowtie ConcertArtist \bowtie temp))$

$\Pi_{aid,\ aname}\sigma_{count > 1}\ temp2$

VI. temp ← $\Pi_{aid,\ aname}\sigma_{vcity = "New\ York"}\ (Concert \bowtie Venue \bowtie ConcertArtist \bowtie Artist)$

result = $\Pi_{aid,\ aname}$ (Artist) - temp

VII. temp ←

$G_{\ min(ticketcost)\ as\ lowestprice}(\sigma_{genre = "Jazz" \wedge vcity = "Chicago" \wedge Year(cdate) = 2017 \wedge Month(cdate) = "October"}\ (TicketType \bowtie$
$Concert \bowtie ConcertArtist \bowtie ArtistGenre))$

$\Pi_{cid,\ cdate,\ ticketcost}\sigma_{genre = "Jazz" \wedge vcity = "Chicago" \wedge Year(cdate) = 2017 \wedge Month(cdate) = "October" \wedge tic878ketcost = temp.lowestprice}$
$((TicketType \bowtie Concert \bowtie\ \ ConcertArtist \bowtie ArtistGenre) \times temp)$

(e)

I. { t | ∃s ∈ Artist (t[aname] = s[aname] ∧
∃u ∈ ArtistGenre (u[aid] = a[aid] ∧ u[genre] = “Jazz”))}

II. { t | ∃v ∈ Venue (t[vcity] = v[vcity] ∧
∃c ∈ Concert (c[vid] = v[vid] ∧ year(c[cdate] = 2016 ∧
∃s ∈ ConcertArtist(s[cid] = c[cid] ∧
∃u ∈ Artist (u[aid] = s[aid] ∧ u[aname] = “Bruno Mars”))))}

III. Cant express it in TRC or DRC due to the aggregation function.

IV. { t | ∃tt ∈ TicketType (t[cid] = tt[cid] ∧ t[ticketclass] = tt[ticketclass]
∧ t[ticketcost] = tt[ticketcost] ∧
∃c ∈ Concert (c[cid] = tt[cid] ∧
∃s ∈ ConcertArtist(s[cid] = c[cid] ∧
∃u ∈ ArtistGenre (u[aid] = s[aid] ∧ u[genre] = “Jazz” ∧
∀ot ∈ TicketType (ot[ticketcost] <= tt[ticketcost] ∧
∃c2 ∈ Concert (c2[cid] = ot[cid] ∧
∃s2 ∈ ConcertArtist(s2[cid] = c2[cid] ∧
∃u2 ∈ ArtistGenre (u2[aid] = s2[aid] ∧ u2[genre] = “Jazz”))))))))}

V. { t | ∃a1 ∈ Artist (t[aid] = a1[aid] ∧ t[aname] = a1[aname] ∧
∃ca1 ∈ ConcertArtist(ca1[aid] = a1[aid] ∧
∃ca2 ∈ ConcertArtist(ca2[cid] = ca1[cid] ∧
∃a2 ∈ Artist (a2[aid] = ca2[aid] ∧ a2[aname] = “Bruno Mars” ∧ a2[aid] != a1[aid]
∧
∃a3 ∈ Artist (a3[aid] = a1[aid] ∧
∃ca3 ∈ ConcertArtist(ca3[cid] != ca1[cid] ∧
∃ca4 ∈ ConcertArtist(ca4[cid] = ca3[cid] ∧
∃a4 ∈ Artist (a4[aid] = ca4[aid] ∧ a4[aname] = “Bruno Mars” ∧ a4[aid] != a3[aid]
))))))))}

VI. { t | ∃a ∈ Artist (t[aid] = t[aname] ∧ t[aname] = a[aname] ∧
¬ ∃ca ∈ ConcertArtist(ca[aid] = a[aid] ∧
∃c ∈ Concert(ca[cid] = c[cid] ∧
∃v ∈ Venue(c[vid] = v[cid] ∧ v[vcity] = “NewYork”))))}

VII. { t | ∃tt ∈ TicketType (t[cid] = tt[cid] ∧ t[ticketcost] = tt[ticketcost] ∧
∃c ∈ Concert (t[cdate] = c[cdate] ∧ c[cid] = tt[cid] ∧ year(c[cdate]) = 2017 ∧
month(c[cdate]) = “Oct”
∃v ∈ Venue(v[vid] = c[vid] ∧ v[vcity] = “Chicago”
∃s ∈ ConcertArtist(s[cid] = c[cid] ∧
∃u ∈ ArtistGenre (u[aid] = s[aid] ∧ u[genre] = “Jazz” ∧

∀ot ∈ TicketType (ot[ticketcost] >= tt[ticketcost] ∧
∃c2 ∈ Concert (c2[cid] = ot[cid] ∧ year(c2[cdate]) = 2017 ∧ month(c2[cdate]) = “Oct”
∃v2 ∈ Venue(v2[vid] = c2[vid] ∧ v2[vcity] = “Chicago”
∃s2 ∈ ConcertArtist(s2[cid] = c2[cid] ∧
∃u2 ∈ ArtistGenre (u2[aid] = s2[aid] ∧ u2[genre] = “Jazz”
))))))))}

Problem 2:

(a) Babysitter (**bsid**, bsname, bsage, bsgender, bsaddress, bsphone, bslanguage, bsdescrip)

Customer (**cusid**, cusname, cusaddress, cusphone)

Child(**chid**, pgid, chname, chage, chgender, chlanguage)
Foreign key: pgid references Customer.cusid
Assumption : The parent/guardian

Schedule(**bsid**, **slottime**, sstatus)
Foreign key: bsid references Babysitter.bsid
Assumption: slottime is the starttime of the slot. The slot should be one hour range and it starts at integral point (like 1:00 p.m.). Status can be “Available” or “Occupied”.

Request(**rid**, cusid, rdate, rstarttime, rrange)
Foreign key: cusid references Customer.cusid

RequestDetail(**rid**, **chid**)
Foreign key: rid references Request.rid
chid references Child.chid

Booking(**bid**, cusid, bsid, bdate, bstarttime, brange, charge, bstatus, rating)
Foreign key: cusid references Customer.cusid
bsid reference Babysitter.bsid
Assumption: Booking can be made only between one customer and one babysitter and event.

BookingDetail(**bid**, **chid**)
Foreign key: bid references Booking.bid
chid references Child.chid

(b)

I.

```
SELECT bsname
FROM Babysitter NATURAL JOIN Schedule
WHERE (Timestamp(slottime) = ‘2017-09-20T15:00:00’ AND status =
“Available”) AND (Timestamp(slottime) = ‘2017-09-20T16:00:00’
AND status = “Available”) AND (Timestamp(slottime) = ‘2017-09-20T17:00:00’
AND status = “Available”)
```

II.

```
SELECT bsid, COUNT(distinct chid)
FROM Booking NATURAL JOIN BookingDetail
WHERE YEAR(bdate) = 2016 AND bstatus = “completed”
GROUP BY bsid
```

III.

```
WITH Income (bsid, totalincome) as (
        SELECT bsid, SUM(charge)
        FROM Booking
        WHERE bstatus = “completed” AND YEAR(bdate) = 2016
)
SELECT bsid, bname
FROM Income NATURAL JOIN Babysitter
WHERE totalincome = (SELECT MAX(totalincome) FROM Income)
```

IV.

```
SELECT cusid, cname
FROM Customer NATURAL JOIN Booking
WHERE bstatus = “completed” AND cusid NOT IN (
        SELECT cusid
        FROM Booking
        WHERE bstatus = “completed” AND rating >=4
)
GROUP BY cusid cname
HAVING COUNT(*) >= 5
```