Kaitlynn Prescott
CS 442 B
10/31/16

Assignment 2

1. Consider a relational database for an online hotel booking company:

Hotel: hid, hname, haddress, hcity
Guest: gid, gname, gaddress, gcity
Room: hid, roomNo, type, price
Booking: gid, hid, roomNo, FromDate, year, nofDays

Booking: g12345, h5555, 220, Jan05, 2016, 8
is Guest g12345 booked room 220 at hotel h5555 for 8 days starting on January 5th, 2016

* gAddress is guest address

Write Following queries in relational algebra:

a) Return name of guests who have booked a Hotel room ~~of type "suite"~~ in NYC in January 2016

$\pi_{gname}(Guest \bowtie (\sigma_{hcity="NYC"}(Hotel) \bowtie \sigma_{FromDate \geq "01/01" \text{ AND } FromDate \leq "01/31" \text{ AND } Year="2016"}(Booking)))$

b) Return the name of guests who have booked a hotel room of type "suite" in NYC for longer than 30 days.

$\pi_{gname}(Guest \bowtie (\sigma_{hcity="NYC"}(Hotel) \bowtie \sigma_{type="suite"}(Room) \bowtie \sigma_{nofDays>30}(Booking)))$

c) Return name of guests who've never booked a hotel for longer than 3 days → only less than or equal to 3

$\rho(Gid, (\pi_{gid,nofdays}(Booking) / \pi_{nofdays}(\sigma_{nofdays \leq 3}(Booking))))$
$\pi_{gname}(Guest \bowtie Gid)$

d) Return ids of hotels in NYC which were not booked at all in 2015

$\rho(NYCHotels, \sigma_{hcity=NYC}(Hotel) \bowtie Booking)$
$\rho(Booked2015, \sigma_{year=2015}(Booking) \bowtie Hotel)$
$\pi_{hid}(NYCHotels - Booked2015)$

e) Return ids of guests who have booked atleast one room of type "Penthouse Suite" in every hotel in NYC

$\rho(NYC, \sigma_{hcity='NYC'}(Hotel))$

$\rho(Penthouse, \sigma_{type="penthouse\ suite"}(Room) \bowtie Booking)$

$\pi_{gid}(Penthouse / NYC)$,

2. Emp(eid: integer, ename: string, age: integer, salary: real)
Works(eid: integer, did: integer)
Dept(did: integer, budget: real, managerid: integer, dname: string)

a) Find names and ages of each employee in both hardware & software departments

```
SELECT E1.ename, E1.age
FROM Emp E1, Works W1, Dept D1
WHERE E1.eid = W1.eid AND
      W1.did = D1.did AND
      D1.dname = 'Hardware'
INTERSECT
SELECT E2.ename, E2.age
FROM Emp E2, Works W2, Dept D2
WHERE E2.eid = W2.eid AND
      W2.did = D1.did AND
      D2.dname = 'Software' ;
```

b) Find ids of Managers who control largest amount $

```
SELECT Managerid
FROM Dept
WHERE budget = MAX(budget);
```

c) Find ids of Managers who manage only departments w/ budget at least $1 million

```
SELECT managerid
FROM Dept
WHERE budget >= 1,000,000 AND
      NOT EXISTS (SELECT managerid
                  FROM Dept
                  WHERE budget < 1,000,000);
```

d) Find names of all employees whose salary exceeds budget of all departments they work in.

```
SELECT E.ename
FROM Emp E
WHERE E.salary > SUM(SELECT D.budget
                     FROM Dept D
                     WHERE E.eid = D.managerid);
```

e) Find name(s) of managers whose department has largest budget

```
SELECT E.ename
FROM Emp E, Dept D
WHERE E.eid = D.managerid AND
      budget = MAX(budget);
```

3. House (id, asking-price, address, postal-code, bath, beds, sq ft, SellerID)
Seller(id, name, home_phone, email, agentID)
Buyer(id, name, home_phone, email, agentID)
Agent(id, name, mobile_phone, email)
Sold(house_id, buyer_id, sale_date, selling_price)

a) Relational Algebra

i) return id, addresses, asking_price & selling_price of all houses whose selling price was lower than asking

ρ(sell_low, $\sigma_{\text{sold.selling\_price} < \text{House.asking\_price}}$(House $\bowtie$ sold))

$\Pi_{\text{house\_id, address, asking\_price, selling\_price}}$(sell_low)

ii) Return names of potential buyers.

ρ(Bought, $\sigma_{\text{Buyer.id} = \text{sold.buyer\_id}}$(sold $\bowtie$ Buyer))

$\Pi_{\text{Buyer.name}}$(Buyer − Bought)

b) SQL

i) For each postal code w/ 5+ houses sold in 2015, Find postal code & average selling price

```
SELECT H.postal_code, AVG(S.selling_price) AS average
FROM House H, Sold S
GROUP BY H.postal_code
HAVING COUNT(*) > 4 ;
```

ii) Find addresses & asking prices of all houses that have minimum 4 bed, 2 bath, not sold.

```
SELECT DISTINCT H.address, DISTINCT H.asking_price
FROM House H
WHERE H.beds >= 4 AND H.baths >= 2 AND
      H.id NOT IN (SELECT S.house_id
                   FROM Sold S );
```

I pledge my honor that I have abided by the Stevens Honor System.