6 2018 SQL Final

This exam was given at the masters level in 2018 for a course just covering SQL. Students had two hours to complete it.

The information below comes from an insurance company's database, which operates under the following model: Agents sell insurance to *households*. A household may purchase multiple types of insurance (e.g. fire, car, earthquake). If a household makes a claim on a policy, then the insurance company can either pay the claim or fight it. If the insurance company fights a claim, then there are expenses associated with it. This database contains only active policies.

- Columns with the same name can be assumed to be the same.
- **Unless stated above, all values are not null.**
- Agent Table:
  - This table contains information regarding the agents in the company.
  - **AgentID:** This the ID of the agent, it is unique to each row. (int)
  - **St:** This is the agent's state. NULL values mean that the agent is international. (string)
  - **Name:** This is the agent's name.
  - **Bonus:** This is equal to "H" or "L" for "High" or "Low" bonus structure. (string)
- Household table:
  - This table contains information about the households that are insured.
  - **HHID:** The unique ID for the household, it is unique to each row. (int)
  - **AgentID:** This is the ID of the agent who manages the household. (int)
  - **Name:** This is the head of household's name. (string)
  - **MultiFam:** Are there multiple people in the household (1 = "Yes", 0 = "No"). (int)
  - **zip:** The zip for the household. (string)
- Policy table:
  - This table contains information on each policy. A household may *multiple* policies.
  - **PolicyID:** The unique ID for each policy. You can assume it is unique to each row. (int)
  - **HHID:** The unique ID for the household. (int)
  - **PolicyType:** The type of insurance policy. (string)
  - **EndDate:** The date the policy expires. (date)
  - **Cost**: The cost charged for the policy. (float)
- Claims Table:
  - This table contains information on claims made against insurance policies. A policy *may* have multiple claims or *no* claims against it.
  - **ClaimID:** The ID associated with the claim, it is unique to each row. (int)
  - **PolicyID:** The policy the claim is against. (int)

  - **Amt:** The amount of money being asked by the policy holder for the claim. (float)
  - **dt:** This is the date that the claim was received. (date)

- Expense Table:
  - This table contains information on legal expenses for disputing a claim. A single claim may have multiple expenses.
  - **EID:** Expense ID, assume it is unique for each row. (int)
  - **ClaimID:** The claim that the expense is against. (int)
  - **EAmt:** The amount of the expense. (float)
  - **dt:** This is the ID of the salesperson. (int)
  - **type:** The type of expense (legal, private investigator, etc.). (float)

Table D.6: *Agent* Table, 228 Rows

AgentID	ST	Name	Bonus
1	CA	Miles Quaritch	H
2	PA	Jake Sully	L
3		Parker Selfridge	L
4	CA	Neytiri	H

Table D.7: *Household* Table, 4,525 Rows

HHID	AgentID	Name	Zip	MultiFam
1	1	John Smith	11217	1
2	1	James McWright	99924	1
3	37	Elrod Lee	12345	0

Table D.8: *Policy* Table, 5,587 Rows

PolicyID	HHID	PolicyType	EndDate	Cost
1	1	Fire	01-01-2019	1,245.76
2	1	Car	01-01-2019	2,247.05
3	5	Flood	07-08-2020	287.56
4	37	Earthquake	03-01-2019	22,476.18

Table D.9: *Claims* Table, 1,225 Rows

ClaimID	PolicyID	Amt	dt
1	22	254.85	09-03-2018
2	22	212.87	09-05-2018
3	188	12,285.96	07-07-2017

Table D.10: *Expense* Table, 2,258 Rows

EID	EAmt	ClaimID	dt	type
1	65.73	1	01-03-2011	Legal
2	75.73	12	02-03-2011	Private Invest.
3	265.04	17	01-05-2011	Filing Fees
4	554.36	8	08-27-2011	Court Fees
5	18.18	22	08-11-2011	Legal

For the following questions write a query that returns the answer **and ONLY the answer.** All questions should be answered with a single SQL query, unless stated otherwise. Do not use CTE's.

1. What are the top five agents (AgentID) in terms of *number* of Households sold to?

```
select AgentID
from Household
group by 1
order by sum(1) desc
limit 5;
```

2. How many Agent's have sold policy's to households in zipcode 11217?

```
select
        count(distinct agentID)
from
        Household
where zipcode = 11217;
```

3. Write a query which returns four columns. The first column should be the year (int), the second should be the month (int) that a policy expires, the third column should be the total costs of all policies that expire during that month-year and the final column should be the number of "Fire" policies that expire that month.

```
select
        date_part('year', EndDate) as yr
        , date_part('month', EndDate) as mnth
        , sum( Cost) as totalCost
        , sum( case when PolicyType = 'Fire' then 1else 0 end) as numberFire
from
        Policy
group by 1,2;
```

4. For each household (HHID), return (a) the HHID, (b) the number of policies it has (c) the total costs associated with those policies.

```
select
        hhid, count(1), sum(Cost)
from
        policy
group by 1;
```

5. Similar to the above question, return the (a) HHID, (b) the number of policies for that household and (c) the number of claims for that household.

```
select
        lhs.hhid, count(distinct rhs1.policyID), count(distinct rhs2.claimID)
from
        household as lhs
left join
        policy as rhs1
        using(hhid)
left join
        claims as rhs2
        using(policyid)
group by 1;
```

6. International Agents need to pay an additional tax on their policy's of 7.5% of the cost. For those policies which expire in 2019, what is the total international tax bill?

```
select
        sum(cost) *.075 as taxpaid
from
        agent
left join
        household
        using( agentID)
left join
        policy
        using( policyID)
where agent.st is null
and date_part('year', enddate) = 2019;
```

7. Write a query which returns the most common policy type (e.g. "Fire") for MultiFam households. Note that most common is the number of policies, NOT cost.

```
select
        policy.policytype
from
        household
        using( agentID)
left join
        policy
        using( policyID)
where multifam = 1
group by 1
order by count(1) desc. limit 1;
```

8. Write a query which returns two rows and two columns. The first column should be "Bonus" type ("H" or "L") and the second should be the number of agents who have that bonus type (number of rows).

```
select bonus, count(1)
from agent group by 1;
```

9. Write a query which returns one rows and two columns, which is the transpose of the previous

question. The first column should be "HighBonus" and contain the number of agents who have Bonus type of High and the second column should be the number of agents who have a Bonus type of Low ("LowBonus")

```
select
        sum( case when bonus = 'H' then 1 else 0 end) as HighBonus
        , sum( case when bonus = 'H' then 1 else 0 end) as LowBonus
from agent;
```

10. What is the total claim (*not policy cost*) amount by policy type?

```
select
        policytype
        , sum(amt) as claimamt
from
        policy
left join
        claims
        using(policyID)
group by 1
```

11. The company is interested in learning if policies from "H" bonus agents have more claims than those of "L" bonus agent. Please write query which returns the total policy costs as well as the total amount paid for claims, by Agent bonus type. This should return two rows and three columns (bonustype, policycosts, claimamt).

```
select
        bonus
        , sum( policy.cost) as policycost
        , sum( rhs2.amt) as claimamt
from
        agent
left join
        household
        using(AgentID)
left join
        policy
        using(HHID)
left join
        (select sum(amt) as amt, policyid from claims group by 2) as rhs2
        using(policyID)
group by 1;
```

12. Write a query which returns the following information: (a) AgentID, (b) The number of policies that they are in charge of, (c) the total number of claims against those policies and (d) the percentage of policies that the agent is in charge of that have a claim against it. Specifically, if a policy has two claims against it, then it should only be counted once in the numerator.

```
select
        AgentID
        , count(distinct policyID) as numPolicies
        , count(distinct claimID) as numclaims
        , count(distinct claimID)::float / count(distinct policyID) as pct
from
        household
left join
        policy
        using( HHID)
left join
        claims
        using( PolicyID)
group by 1;
```

13. Write a query which has three columns: year of policy expiration (as an integer), month of policy expiration (as an integer) and a running sum of the dollar amount of policies expiring over time. There should be one row for each year-month combination.

```
select yr, mon
        , sum( cost) over( order by yr asc, mon asc
                  rows between unbounded preceding and current row)
from
(select
        date_trunc('year', enddate) as yr
        , date_trunc('month', enddate) as mon
        sum( cost) as cost
from
        policy
group by 1,2) as iq
```

14. We define a "house" risk zip code to be one where the total cost of "Fire", "Flood" and "Earthquake" is more than twice as large as the cost of "Car" policies within that zip code. What percentage of zip codes are high risk (return a single number, the percent of zip codes are "house" risk)? You may assume that there is at least one policy of each policy type within each zip code.

```
select
        avg( case when 2* hr > car then 1 else 0 end)  as pct
from
        (select
                zip
                , sum( case when PolicyType in ('Fire', 'Earthquake', 'Flood)
                        then cost else 0 end) as hr
                , sum( case when PolicyType = 'Car'
                        then cost else 0 end) as car
        from
                household
        left join
                policy
        using(policyID)
        group by 1) as innerq;
```

15. Write a query which returns the total expense amount of each "type" of Expense as well as the type of expense, making sure to sort from the largest total amount to the smallest total amount.

```
select
        type
        , sum(Eamt) as totalamt
from
        expenses
group by 1
order by 2 desc;
```

16. Write a query which returns the state with the largest number of "H" bonus agents. This should return 1 row and 1 column

```
select
        st
from
        agent
        where bonus = 'H'
group by 1
order by count(1) desc limit 1;
```

17. Of all the agents (agentID) who have ever sold a policy to zip code to 11217, which one had the highest average policy cost over all their policies that they sold?

```
select household.agentID
from
        household
left join
        policy
using( hhid)
where agentid in
        (select distinct agentid from household where zip = 11217)
group by agentID
order by avg( cost) desc limit 1;
```

18. Which household (HHID) has the most number of policies (assume that this is unique and that there exists a household with more policies than any other household)

```
select
        HHID
from
        policy
group by HHID
order by count(1) desc limit 1;
```

19. We are interested in the cash flow associated with claims, by policy type. Write a query which returns the 3 day moving average of the total claims (amt) received *per day* by policy type as well as the day-over-day percent change. There should be one row per day-policy type combination and four columns in the table (dt, policy type, moving average and percent change). To compute the percent

change, take today's total amount and divide by yesterday's. If today is 1/5/2018, then the three days in the moving average should be 1/3, 1/4 and 1/5. You can assume that every day of interest is represented in the policy table for all policies and that there are no days without claims for every policy type.

```
select
        dt, policytype
        , avg( amt) over( partition by policytype
                order by dt asc rows between 2 preceding and current row)
        , amt / lag( amt) over( partition by policytype
                order by dt asc )
from
(select
        sum( amt) as amt
        , dt
        , policytype
from
        policy
left join
        claims
        using( PolicyID)
        group by 2,3) as IQ
```

20. What month (just month, e.g. 12 for "December") is the most common end date for policies (by number of policies)?

```
select date_part('month', enddate)
from policy group by 1 order by count(1) desc limit 1;
```

21. It turns out that California Agents (those with state = "CA") were lying on costs associated with "Fire" Policies. In particular, every policy which was more than $500 had an additional $100 of fraud added to it. Write a query which returns two columns and one row. The first column should be the number of affected policies and the second should be the total amount (in dollars) of fraud.

```
select
        sum(1) as fclaims, 100 * sum(1) as dollarsfraud
from
        (select HHID, policyID, cost from policy
                where policytype = 'Fire' and cost > 1000) as lhs
left join
        household
        using(HHID)
where agentID in (select agentID from agent where st = 'CA') ;
```

22. What percentage of expenses have a claim amount associated with them larger than $1000? Return a single number which is this percent.

```
select
        sum( case when claims.amt > 1000 then 1 else 0 end) / sum(1) as pct
from
        expense
left join
        claims
        using(claimID);
```

23. Write a query which returns three columns. The first column should be policy type, the second should be the number of Multifam households which purchased that policy type and the third should be the number of NON Multifam households (Multifam = 0) that purchased it. There should be one row per policy type.

```
select
        policytype
        , sum( case when multifam = 1 then 1 else 0 end)
        , sum( case when multifam = 0 then 1 else 0 end)
from household
left join policy using( HHID )
group by 1;
```

24. Which policy type has the highest average cost? Return one row and one column.

```
select policytype
from policy
group by 1
order by avg( cost) desc
limit 1;
```

25. Agent's are paid based on the following formula: If they have a "H" type bonus plan, then they receive $100 for each policy they sell + 10% of the cost of that policy. If an agent has a low type ("L") bonus plan, they receive $200 for each policy they sell + 5% of the cost of the policy. Compute the total wages paid to each agent. This should return two columns: agentID and their wages.

```
select
        agentID
        , case
                when bonus = 'H' then 100 * numpolicies + .1 * totalcost
                else 200 * numpolicies + .05 * totalcost
from
        (select
                agentID
                , max( bonus ) as bonus
                , count(1) as numpolicies
                , sum( cost) as totalcost
        from
                agent
                left join
                household
                using( agentID)
        left join
                policy
                using(HHID)
        group by 1) as innerQ
```

26. **Bonus**: All the agent's in the agent table are from a movie. What movie?