Insurance Database SQL Queries Report

Gustavo A. Fernandez

Introduction:

This project aims to showcase SQL querying capabilities using a fictional insurance database. The database consists of tables related to people, cars, accidents, and relationships. Understanding and querying such a database is essential for any data analyst or database administrator working in the insurance domain.

Database Schema:

person(driver-id, name, address)

car(car-license, model, year)

accident(report-number, date, location)

owns(drive-id, car-license)

participated(driver-id, car-license, report-number, damage amount)

Queries:

Task:Find the total number of people who owned cars that were involved in accidents in 1989.

SQL Query:

```
select *
from owns
natural join participated natural join accident
where date like '%1989%'
```

Task:Add a new accident to the database; assume any values for required attributes.

SQL Query:

```
insert into accident
values (2,'25 Jan 2021','Maryland’)
```

Task:Delete the Mazda belonging to “John Smith”.

SQL Query:

```
delete
from car
where model like '%Mazda%' and car_license in (
```

Task:Find the names of all employees who work for First Bank Corporation.

SQL Query:

```
select employee.employee_name
from employee
natural join works JOIN company on
works.company_id=company.company_id
where company_name='First Bank'
```

Task:Find the names and cities of residence of all employees who work for First Bank Corporation.

SQL Query:

```
select employee.employee_name, employee.city
from employee
natural join works JOIN company on
works.company_id=company.company_id
where company_name='First Bank'
```

Task:Find all employees who earn more than the average salary of all employees of their company.

SQL Query:

```
select employee_name
from(
```

Task:Find the company that has the smallest payroll.

SQL Query:

```
select company_name
from (
select company_name,AVG(salary) as AvgSal2
from employee,works,company
where employee.employee_id=works.employee_id and
works.company_id=company.company_id
group by company_name
) as T5
```

Task:1.4. Delete the Mazda belonging to “John Smith”.

SQL Query:

```
delete
from car
where model like '%Mazda%' and car_license in (
select car_license
from owns
natural join person
where name='John Smith'
)
```

Task:1.5. Update the damage amount for the car with license number “AABB2000” in the accident with report number “AR2197” to $3000.

SQL Query:

```
update participated
set damage_amount=3000
where report_number='AR2197'
```

2. Employee Database:

Figure 2. Employee database

employee(employee-id, employee-name, street, city)

works(employee-id, company-id, salary)

company(company-id, company-name, city)

manages(employee-id, manager-id)

Task:2.1. Find the names of all employees who work for First Bank Corporation.

SQL Query:

```
select employee.employee_name
from employee
natural join works JOIN company on
works.company_id=company.company_id
```

```
where company_name='First Bank'
```

Task:2.12. Find those companies whose employees earn a higher salary, on average, than the average of First Bank Corporation.

SQL Query:

```
select distinct(company_name)
from(
select company_name, salary,AVG(salary) over (partition by
company_name) as AverageSalary
from employee,works,company
where employee.employee_id=works.employee_id and
works.company_id=company.company_id
) as T
where AverageSalary > (
                        select AVG(salary) from Works,company
                        where company_name='First Bank'
                        and works.company_id=company.company_id
                         )
```

Conclusion:

Through this project, we have explored various SQL queries that help extract meaningful insights from the insurance database. Constructing and executing these queries efficiently is crucial for making informed decisions. Future improvements could involve optimizing query performance, incorporating more complex SQL functions, or expanding the database schema to capture more granular details about each entity.