

**SCHOOL OF ENGINEERING AND TECHNOLOGY**

**COURSEWORK FOR**

BSC (HONS) IN COMPUTER SCIENCE

BSC (HONS) INFORMATION TECHNOLOGY BACHELOR OF SOFTWARE ENGINEERING (HONS)

BSC (HONS) INFORMATION SYSTEMS (DATA ANALYTICS)

BSC (HONS) INFORMATION TECHNOLOGY (COMPUTER NETWORKING AND SECURITY)

**YEAR 1;** ACADEMIC SESSION AUGUST 2022

**SEG1201: DATABASE FUNDAMENTALS**

**PRESENTATION VIDEO (09:51) :** [**https://youtu.be/RNIixln1Ols**](https://youtu.be/RNIixln1Ols)

**Alternatively, if above link has issue, do access this link for video:** [**https://imailsunwayedu-my.sharepoint.com/:f:/g/personal/20024253\_imail\_sunway\_edu\_my/EthxqUI5IcFBhvIdbvCMQ-YB4tP3Xlv7EfRlGC\_JjMoB8A?e=u6Ckor**](https://imailsunwayedu-my.sharepoint.com/:f:/g/personal/20024253_imail_sunway_edu_my/EthxqUI5IcFBhvIdbvCMQ-YB4tP3Xlv7EfRlGC_JjMoB8A?e=u6Ckor)

**PRESENTATION VIDEO (09:51) :** [**https://youtu.be/RNIixln1Ols**](https://youtu.be/RNIixln1Ols)

**Alternatively, if above link has issue, do access this link for video:** [**https://imailsunwayedu-my.sharepoint.com/:f:/g/personal/20024253\_imail\_sunway\_edu\_my/EthxqUI5IcFBhvIdbvCMQ-YB4tP3Xlv7EfRlGC\_JjMoB8A?e=u6Ckor**](https://imailsunwayedu-my.sharepoint.com/:f:/g/personal/20024253_imail_sunway_edu_my/EthxqUI5IcFBhvIdbvCMQ-YB4tP3Xlv7EfRlGC_JjMoB8A?e=u6Ckor)

**Scenario 1: SUNNY Insurance Company**

SUNNY Insurance Company is a company that offers health and investment-linked life insurance packages for the customers/clients. The company has several branches which each of them has branch number, branch name, branch location and number of agents.

A branch hires at least one agent, but any agent must work under a branch. An agent who is working for a branch under the company will be given an agent number. The agents at SUNNY Insurance Company have their personal particulars including agent name, street, city, state, zip code and contact number stored on database.

An agent always linked to the customers of the company. However, a customer deals with an agent only. For each customer, the company records the customer number, customer name, gender, race, street, city, state, postcode, and mobile number. A customer may buy more than one insurance policy and similarly, the company insurance policies are designed for all level of customers. A policy application has arrived for approving each policy that has chosen by the customers. By doing this, each policy application approved by the company will be given a policy date that means the policy is valid onwards from the assigned date.

Each customer policy has the records of his policy number, owner name, plan, sum assured, and assured name. Besides of having the protection, the policy owners have the benefit of saving money if the policy is within the policy period. A policy has four types of plans, but any plan features at most one policy. Policy plan records the plan, policy period, monthly saving, and annual premium.

To keep records of customer’s saving amount, the company stores them under their policy

account. A policy may have a policy account, but a policy account must arrive with a policy. Each policy is assigned to a policy account that records the account number, frequency, and current balance.

**PART 1: ENHANCED SCENARIO**

**Enhanced scenario:**

The company monitors payment made by each customer by recording the payment amount, payment number, payment description and date. A customer can make many payments, but each payment comes from one customer only.

To keep track of each customer's annual premium, the company records their policy number, payment number and year of payment.

When customers need to claim their health insurance, a payout will be claimed by the policy owner. A payout will record policy owner’s payout number, policy number, assured name, description, and date of payout. A policy owner can claim many payouts and each payout can be claimed by one policy owner only.

For customers that own a policy account, the company records their savings claims by their account number, claim amount and date. A policy account can make many claims, but each claim can only come from one policy account. (143 words)

**Business rules:**

A branch can hire many agents.

Each agent is hired by only one branch.

An agent consults with many customers.

Each customer consults with one agent only.

Each customer can register in more than one policy application.

Each policy application is filled by one customer only.

Each customer policy derives from many policies applications.

Each policy application produces one customer policy only.

Each policy plan is featured in many customers policy.

Each customer policy features at least one policy plan only.

Each customer policy may have a policy account.

Some policy owners do not have a policy account.

Each policy account belongs to only one policy owner.

Each policy account can make many or no savings claim.

Each savings claim can be made by one policy account.

Each customer policy can make many or no payouts.

Each payout is made by one customer policy only.

Each customer policy can make many annual premiums.

Each annual premium made by one customer policy only.

Each payment can be made for many annual premiums.

Each annual premium can only come from one payment.

Each customer can make many payments.

Each payment is made by one customer only.

**PART 2: DESIGINING DATABASE**

1. **Initial entity relationship diagram (ERD):**

A picture containing text, white

Description automatically generated

1. **Translate initial ERD to relational database model (RDM):**

**Bold: Primary key**

Underline: Foreign key

AGENT (**agent number**, branch number, agent name, street, city, state, zip code, contact number)

ANNUAL\_PREMIUM (**policy number, payment number**, amount, year)

BRANCHES (**branch number**, branch name, branch location, number of agents)

CUSTOMER (**customer number**, agent number, customer name, gender, race, street, city, state, postcode, mobile number)

CUSTOMER\_POLICY (**policy number**, plan ,owner name, sum assured, assured name)

PAYMENT (**payment number**, customer number, payment description, amount, date)

PAYOUT (**payout number**, policy number, assured name, amount, payout date, payout description)

POLICY\_ACCOUNT (**account number**, policy number, frequency, current balance)

POLICY\_APPLICATION (**policy number, customer number**, policy date)

POLICY\_PLAN (**plan**, policy period, monthly saving, annual premium)

SAVINGS\_CLAIM (**claim number**, claim amount, claim date, account number)

**Attribute description table:**

|  |  |
| --- | --- |
| Attribute name | Attribute description |
| BRANCHES | Company’s different branch details |
| AGENT | Agents of the company that works under a branch. |
| CUSTOMER | Customer that purchases the insurance policy. |
| CUSTOMER\_POLICY | Insurance policy that is purchased by the customer for the assured (can be customer themselves or different person) |
| POLICY\_APPLICATION | Keeps track of approved policy application. A policy application is approved by giving a policy date that means policy is valid onwards from assigned date. |
| POLICY\_PLAN | Keeps information of the four insurance policy plan that can be selected by customer. |
| POLICY\_ACCOUNT | Keeps track of savings amount, only for customers that wants to keep savings. |
| PAYMENT | The amount of money that the insurance company receives from a customer. |
| PAYOUT | Keeps track of sum of money claimed by customer for health insurance purpose. |
| SAVINGS\_CLAIM | Keeps track of sum of money claimed by policy account owners. |
| ANNUAL\_PREMIUM | Keeps track of annual premium payment from customers because each policy has annual premium that needs to be paid every year. |

1. Normalize initial **RDM**

AGENT (**agent number** , agent name, street, city, state, zip code, contact number)

AGENT-BRANCH (**agent number**, branch number)

ANNUAL\_PREMIUM (**policy number, payment number**, amount, year)

BRANCHES (**branch number**, branch name, branch location, number of agents)

CUST\_POLICY\_ASSURED (**policy number**, sum assured, assured name)

CUST\_POLICY\_OWNER (**policy number**, owner name)

CUST\_POLICY\_PLAN (**policy number** plan)

CUST-AGENT (**customer number**, agent number)

CUSTOMER (**customer number**, customer name, gender, race, street, city, state, postcode, mobile number)

PAYMENT (**payment number**, customer number)

PAYMENT-CUST (**payment number**, customer name, payment description, amount, date)

PAYOUT (**payout number**, policy number)

PAYOUT-DETAILS (**payout number**, assured name, amount, payout date, payout description)

POLICY\_ACC\_CUST (**account number**, policy number)

POLICY\_ACC-BALANCE (**account number**, frequency, current balance)

POLICY\_APPLICATION (**policy number, customer number**, policy date)

POLICY\_PLAN (**plan**, policy period, monthly saving, annual premium)

SAVINGS\_CLAIM (**claim number**, account number)

SAVINGS\_CLAIM\_AMOUNT(**claim number**, claim amount, claim date)

1. Finalized ERD after normalizing.

Diagram

Description automatically generated

**Part 3: Implement our database**

**SQL SCRIPT:** [**https://imailsunwayedu-my.sharepoint.com/:f:/g/personal/20024253\_imail\_sunway\_edu\_my/EtlhkV0XUJxHrsi\_m\_Gt8jIBFsdCXTuDVdhR0snVzuB6dw?e=Uj5NVj**](https://imailsunwayedu-my.sharepoint.com/:f:/g/personal/20024253_imail_sunway_edu_my/EtlhkV0XUJxHrsi_m_Gt8jIBFsdCXTuDVdhR0snVzuB6dw?e=Uj5NVj)

**Part 4: Query a database**

Sunny Insurance would like to reduce the next annual premium amount for customers that has paid their annual premium and has not claimed payout since. However, customers that has not paid annual premium since application on time are not eligible for annual premium reduction. Therefore, the company needs to produce a report for customers that is eligible for the annual premium reduction to inform them.

To notify eligible customer on annual premium reduction, each SQL statement must produce some results.

1. Write a user query with set operation and a GROUP BY clause to retrieve the policy number of customers that deserves annual reduction along with customer name and number.

select p.policy\_payment as policy\_id\_annual\_reduction, c.customername, c.customerno

FROM customer c, payment p, annual\_premium ap

WHERE p.payment\_num=ap.payment\_num\_annual\_premium

AND c.customerno=p.customer\_payment

MINUS

select p.policy\_num\_payout, c.customername, c.customerno

from payout p , customer c, policy\_application pa

where pa.customer\_num\_application= c.customerno

AND p.policy\_num\_payout=pa.policy\_num\_application

group by p.policy\_num\_payout, c.customername, c.customerno

having count(\*)>0;

Graphical user interface, application, Teams

Description automatically generated

1. Write user query with multiple subqueries, LIKE keyword and ORDER BY clause to retrieve customer number, name and phone number that deserves annual reduction.

Select c.customerno as customer\_id\_for\_annual\_reduction, c.customername, c.mobileno

From customer c

Where c.customerno IN (

Select pa.customer\_num\_application

From policy\_application pa

Where pa.policy\_num\_application IN (

Select p.policy\_payment

From payment p

Where p. Payment\_num IN (

Select pc.paymentno

From payment\_cust pc

Where  pc.description LIKE 'Annual Premium'

AND pc.paymentno IN (

Select p.payment\_num

From payment p

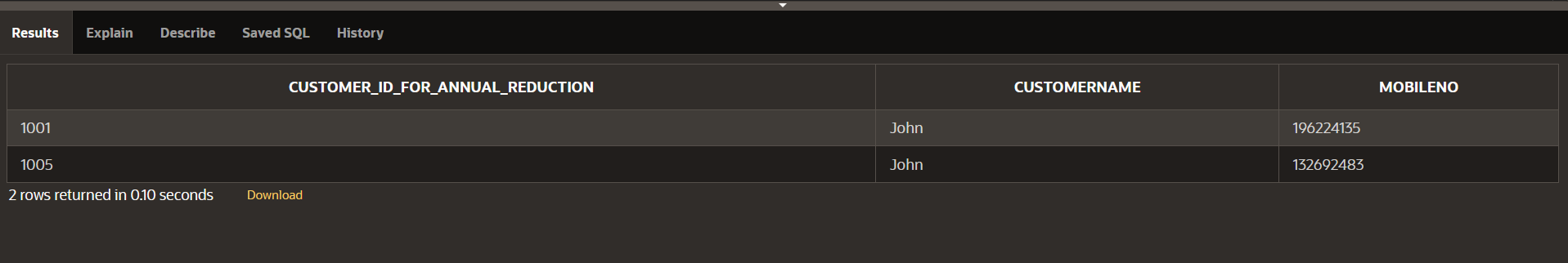
Where p.policy\_payment NOT IN(

Select py.policy\_num\_payout

From payout py

)))))

ORDER BY c.customerno;



1. Write a user query with an outer join and 3 user conditions to extract the names and customer numbers of customers who are eligible for the reduction. One of the conditions uses the LIKE keyword.

SELECT  c.customername, c.customerno, p.policy\_payment as policy\_no

    FROM customer c, payment p, annual\_premium ap

    WHERE c.customerno = p.customer\_payment

    AND p.payment\_num = ap.payment\_num\_annual\_premium

    AND p.policy\_payment IN

    (SELECT p.policy\_payment as policy\_no

        FROM payment p

        LEFT OUTER JOIN payout po ON p.policy\_payment = po.policy\_num\_payout

        WHERE po.payout\_num IS null);

A screenshot of a computer screen

Description automatically generated

1. Write a user query with 4 table aliases and using date functions, to ensure that the customer has paid their annual premium on time. Retrieve the name, customer number and policy number for the eligible customers.

SELECT c.customername, c.mobileno, p.policy\_payment as policy\_no

    FROM customer c, payment p, payment\_cust pc, policy\_application pa

    WHERE c.customerno = p.customer\_payment

    AND p.payment\_num = pc.paymentno

    AND p.policy\_payment = pa.policy\_num\_application

    AND pc.description LIKE 'Annual Premium'

    AND trunc( months\_between( pa.policy\_date, pc.payment\_date ) /12 ) < 1

    AND p.policy\_payment NOT IN(

        Select po.policy\_num\_payout

        From payout po)

    ORDER BY c.customername;

A screenshot of a computer screen

Description automatically generated