

Name: Siming Deng

- Create Insurance tables

1. Claims table

Query Editor Query History

```
1 CREATE TABLE claims (
2     Claim_ID_Number      VARCHAR (100) PRIMARY KEY NOT NULL,
3     Claim_Receipt_Date   DATE,
4     Claim_Description    VARCHAR (255),
5     Claim_Birth_Date     DATE,
6     Law_Suit_Indicator   VARCHAR (10),
7     Claimant_First_Name  VARCHAR (30),
8     Claimant_Last_Name   VARCHAR (50),
9     Policy_Number        INT,
10    Claimant_Occupation  VARCHAR (100),
11    Loss_Date            DATE,
12    Claim_Amount          DECIMAL (30, 2),
13    Claim_Paid_Date       DATE
14 )
15
16 SELECT *
17 FROM claims;
```

Data Output Explain Messages Notifications

claim_id_number	claim_receipt_date	claim_description	claim_birth_date	law_suit_indicator	claimant_first_name	claimant_last_name
[PK] character varying (100)	date	character varying (255)	date	character varying (10)	character varying (30)	character varying(50)

2. Customers table

> claims

> customers

 ↳ Columns (13)

- customer_id_number
- customer_first_name
- customer_last_name
- customer_type
- business_name
- address_street_name_line_one
- address_street_name_line_two
- address_city
- address_state
- address_postal_code
- address_type
- customer_birth_date
- risk_score

> Constraints

> Indexes

> RLS Policies

> Rules

> Triggers

Query Editor Query History

```
1 CREATE TABLE customers (
2     customer_id_number      INT PRIMARY KEY NOT NULL,
3     customer_first_name     VARCHAR (30),
4     customer_last_name      VARCHAR (50),
5     customer_type           VARCHAR (30),
6     business_name           VARCHAR (50),
7     address_street_name_line_one  VARCHAR (50),
8     address_street_name_line_two  VARCHAR (50),
9     address_city             VARCHAR (30),
10    address_state            VARCHAR (30),
11    address_postal_code      INT,
12    address_type              VARCHAR (30),
13    customer_birth_date      DATE,
14    risk_score                INT
15 );
```

Data Output Explain Messages Notifications

CREATE TABLE

Query returned successfully in 53 msec.

3. Policy_details table

The screenshot shows the MySQL Workbench interface. On the left, the database browser tree shows a node for 'policy_details' under 'Tables'. A sub-node 'Columns (8)' is selected, listing columns: customer_id_number, policy_number, policy_effective_date, policy_expiration_date, policy_limit, policy_type, currency_type, and premium_cost. To the right, the 'Query Editor' tab is active, displaying the SQL code for creating the table:

```
1 CREATE TABLE policy_details (
2     customer_id_number INT,
3     policy_number VARCHAR(100) PRIMARY KEY NOT NULL,
4     policy_effective_date DATE,
5     policy_expiration_date DATE,
6     policy_limit DECIMAL (50,2),
7     policy_type VARCHAR (50),
8     currency_type VARCHAR (20),
9     premium_cost DECIMAL (50,2)
10 )
```

Below the query editor, the message 'Query returned successfully in 34 msec.' is displayed.

4. Reserves table

The screenshot shows the MySQL Workbench interface. On the left, the database browser tree shows a node for 'reserves' under 'Tables'. A sub-node 'Columns (5)' is selected, listing columns: customer_id_number, policy_number, reserve_id, reserved_amount, and reserved_date. To the right, the 'Query Editor' tab is active, displaying the SQL code for creating the table:

```
1 CREATE TABLE reserves (
2     customer_id_number INT,
3     policy_number VARCHAR(100),
4     reserve_id serial PRIMARY KEY NOT NULL,
5     reserved_amount DECIMAL (20,2),
6     reserved_date DATE
7 );
```

Below the query editor, the message 'Query returned successfully in 51 msec.' is displayed.

- Add foreign keys

1. The foreign key for the policy_details table

The screenshot shows the database schema for the policy_details table. It includes columns like customer_id_number, policy_number, policy_effective_date, policy_expiration_date, policy_limit, policy_type, currency_type, and premium_cost. A foreign key constraint, fk_customers, is defined on the customer_id_number column, referencing the customer_id_number column in the customers table.

```

ALTER TABLE policy_details
ADD CONSTRAINT fk_customers
FOREIGN KEY (customer_id_number)
REFERENCES customers (customer_id_number);

```

- Insert script for each table

1. Data input for the customers table

The screenshot shows an insert script for the customers table. It inserts three rows of data: one for Mary Roberts (Individual, Residential) and two for Joe Smith and Aditya Reddy (Business).

```

INSERT INTO customers
VALUES
(777543325, 'Mary', 'Roberts', 'Individual', NULL, '123 Main Street', NULL, 'Chicago', 'Illinois', 60007, 'Residential', '3/20/86', 75),
(777543800, 'Joe', 'Smith', 'Individual', NULL, '123 Park Avenue', NULL, 'New York', 'New York', 10017, 'Residential', '2/27/63', 70),
(776983401, 'Aditya', 'Reddy', 'Business', 'ABC Systems', '222 Hyde Street', 'Floor 2', 'San Francisco', 'California', 94123, 'Business', NULL, 50);

```

	customer_id_number	customer_first_name	customer_last_name	customer_type	business_name	address_street_name_line_one
1	777543325	Mary	Roberts	Individual	[null]	123 Main Street
2	777543800	Joe	Smith	Individual	[null]	123 Park Avenue
3	776983401	Aditya	Reddy	Business	ABC Systems	222 Hyde Street

2. Data input for the policy_details table

The screenshot shows an insert script for the policy_details table. It inserts six rows of data, mostly for automobile policies with varying start and end dates and different limits.

```

INSERT INTO policy_details
VALUES
(777543800, 'P987654321', '2/11/20', '2/11/21', 500000.00, 'Automobile', 'USD', 2600.00),
(777543800, 'P987654321', '2/11/19', '2/11/20', 500000.00, 'Automobile', 'USD', 2300.00),
(777543800, 'P987654321', '2/11/18', '2/11/19', 500000.00, 'Automobile', 'USD', 2150.00),
(777543800, 'P987654321', '2/11/17', '2/11/18', 500000.00, 'Automobile', 'USD', 2100.00),
(777543325, 'P987654111', '3/15/20', '3/15/21', 5000000.00, 'Excess/Umbrella', 'USD', 5500.00),
(776983401, 'P987654222', '6/15/20', '6/15/20', 3000000.00, 'Property', 'USD', 10000.00);

```

	customer_id_number	policy_number	policy_effective_date	policy_expiration_date	policy_limit	policy_type	currency_type	premium_cost
1	777543800	P987654321	2020-02-11	2021-02-11	500000.00	Automobile	USD	2600.00
2	777543800	P987654321	2019-02-11	2020-02-11	500000.00	Automobile	USD	2300.00
3	777543800	P987654321	2018-02-11	2019-02-11	500000.00	Automobile	USD	2150.00
4	777543800	P987654321	2017-02-11	2018-02-11	500000.00	Automobile	USD	2100.00
5	777543325	P987654111	2020-03-15	2021-03-15	5000000.00	Excess/Umbrella	USD	5500.00
6	776983401	P987654222	2020-06-15	2020-06-15	3000000.00	Property	USD	10000.00

3. Data input for claims table

Data Output Explain Messages Notifications								
	claim_id_number [PK] character varying (100)	claim_receipt_date date	claim_description character varying (255)	claim_birth_date date	law_suit_indicator character varying (10)	claimant_first_name character varying (30)	claimant_last_name character varying (50)	policy_number character varying (100)
1	C123450000	2020-12-05	Automobile accident	1975-08-01	N	Joe	Smith	P987654321
2	C123441000	2019-08-05	Automobile accident	1975-08-01	N	Joe	Smith	P987654321
3	C123440000	2020-10-15	Robbery	1986-03-20	N	Mary	Roberts	P987654111
4	C345478900	2020-07-13	Property	1990-05-27	N	Aditya	Reddy	P987654222

4. Data input for reserves table

Data Output Explain Messages Notifications				
	customer_id_number integer	policy_number character varying (100)	reserved_amount numeric (20,2)	reserved_date date
1	777543325	P987654111	5000.00	'3/15/20')
2	776983401	P987654222	7000.00	'6/15/20')
3	777543800	P987654321	1500.00	'2/11/17')
4	777543800	P987654321	1500.00	'2/11/18')
5	777543800	P987654321	1500.00	'2/11/19')

- 5 SQL to extract information from tables

1. ID and Name of the claimants who filed a loss amount for more than \$5,000.

Query Editor Query History

```
1 SELECT claim_id_number, claimant_first_name, claimant_last_name, claim_amount
2 FROM claims
3 WHERE claim_amount > 5000;
```

Data Output Explain Messages Notifications

	claim_id_number [PK] character varying (100)	claimant_first_name character varying (30)	claimant_last_name character varying (50)	claim_amount numeric (30,2)
1	C123441000	Joe	Smith	7600.00
2	C123440000	Mary	Roberts	25000.00
3	C345478900	Aditya	Reddy	10000.00

2. The total amount of the claims paid.

Query Editor Query History

```
1 SELECT SUM(claim_paid_amount) AS total_claims_paid
2 FROM claims;
```

Data Output Explain Messages Notifications

	total_claims_paid numeric
1	35100.00

3. Name and ID of the claimant who got the highest claim paid.

Query Editor Query History

```
1 SELECT claim_id_number, claimant_first_name, claimant_last_name, claim_paid_amount
2 FROM claims
3 WHERE claim_paid_amount IS NOT NULL
4 ORDER BY claim_paid_amount DESC
5 LIMIT 1;
```

Data Output Explain Messages Notifications

	claim_id_number [PK] character varying (100)	claimant_first_name character varying (30)	claimant_last_name character varying (50)	claim_paid_amount numeric (20,2)
1	C123440000	Mary	Roberts	20000.00

4. The total amount of money the insurance company has set aside to cover claims.

Query Editor Query History

```
1 SELECT SUM(reserved_amount) AS total_reserved_amount
2 FROM reserves;
```

Data Output Explain Messages Notifications

	total_reserved_amount numeric
1	16500.00

5. Find the history of policy effective date and policy expiration date for Joe Smith.

Query Editor Query History

```
1 SELECT p.policy_effective_date,
2       p.policy_expiration_date,
3       c.customer_id_number,
4       c.customer_first_name,
5       c.customer_last_name
6 FROM customers as c
7 JOIN policy_details as p ON c.customer_id_number = p.customer_id_number
8 WHERE c.customer_first_name = 'Joe' AND c.customer_last_name = 'Smith';
```

Data Output Explain Messages Notifications

	policy_effective_date date	policy_expiration_date date	customer_id_number integer	customer_first_name character varying (30)	customer_last_name character varying (50)
1	2020-02-11	2021-02-11	777543800	Joe	Smith
2	2019-02-11	2020-02-11	777543800	Joe	Smith
3	2018-02-11	2019-02-11	777543800	Joe	Smith
4	2017-02-11	2018-02-11	777543800	Joe	Smith