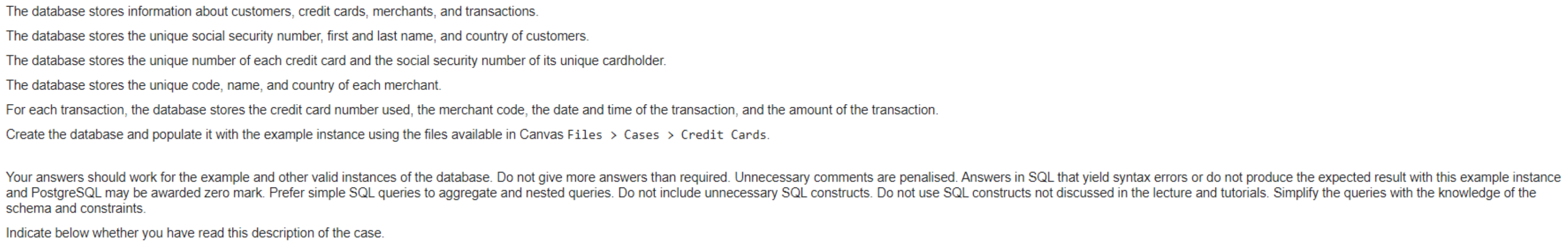
## Creation of Database:



The database stores information about customers, credit cards, merchants, and transactions.

The database stores the unique social security number, first and last name, and country of customers.

The database stores the unique number of each credit card and the social security number of its unique cardholder.

The database stores the unique code, name, and country of each merchant.

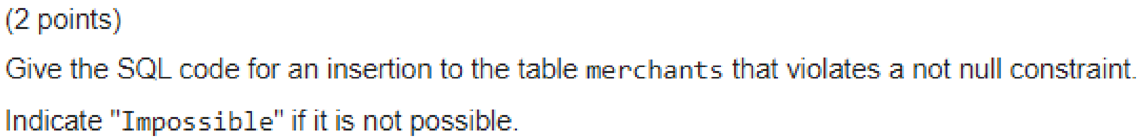
For each transaction, the database stores the credit card number used, the merchant code, the date and time of the transaction, and the amount of the transaction.

Create the database and populate it with the example instance using the files available in Canvas `Files > Cases > Credit Cards`.

Your answers should work for the example and other valid instances of the database. Do not give more answers than required. Unnecessary comments are penalised. Answers in SQL that yield syntax errors or do not produce the expected result with this example instance and PostgreSQL may be awarded zero mark. Prefer simple SQL queries to aggregate and nested queries. Do not include unnecessary SQL constructs. Do not use SQL constructs not discussed in the lecture and tutorials. Simplify the queries with the knowledge of the schema and constraints.

Indicate below whether you have read this description of the case.

## Query 1:

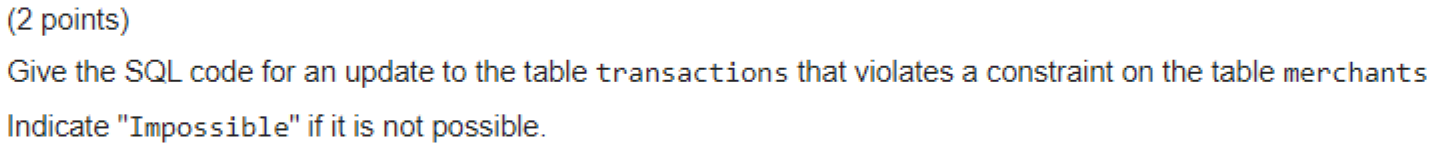


(2 points)

Give the SQL code for an insertion to the table `merchants` that violates a not null constraint.

Indicate "`Impossible`" if it is not possible.

## Query 2:

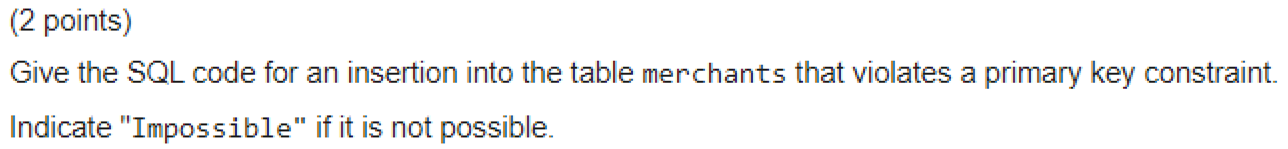


(2 points)

Give the SQL code for an update to the table `transactions` that violates a constraint on the table `merchants`

Indicate "`Impossible`" if it is not possible.

## Query 3:

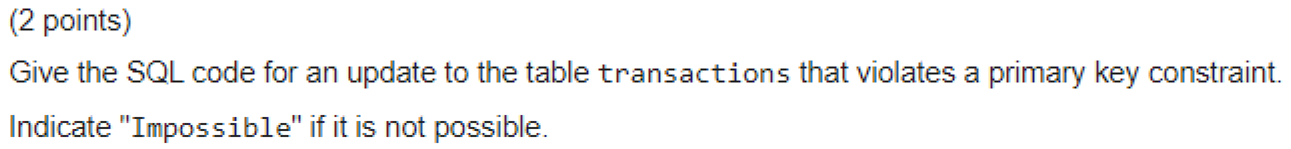


(2 points)

Give the SQL code for an insertion into the table `merchants` that violates a primary key constraint.

Indicate "`Impossible`" if it is not possible.

## Query 4:

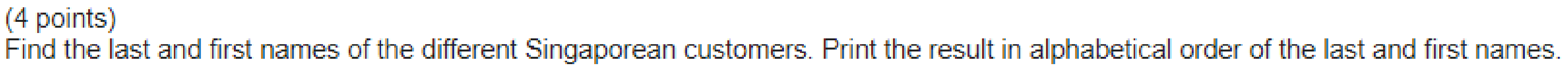


(2 points)

Give the SQL code for an update to the table `transactions` that violates a primary key constraint.

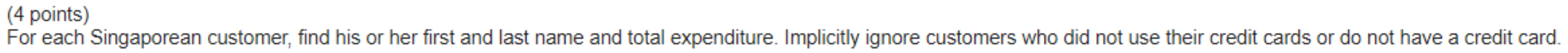
Indicate "`Impossible`" if it is not possible.

## Query 5:



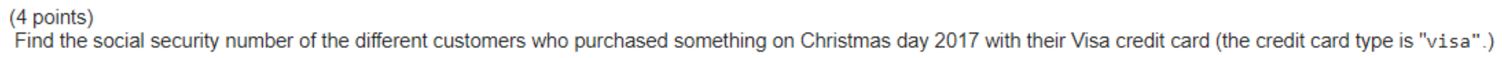
(4 points)
Find the last and first names of the different Singaporean customers. Print the result in alphabetical order of the last and first names.

## Query 6:



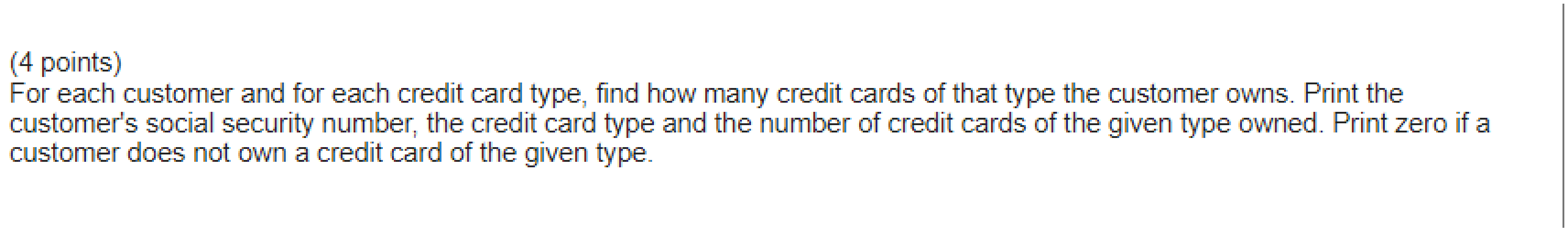
(4 points)
For each Singaporean customer, find his or her first and last name and total expenditure. Implicitly ignore customers who did not use their credit cards or do not have a credit card.

## Query 7



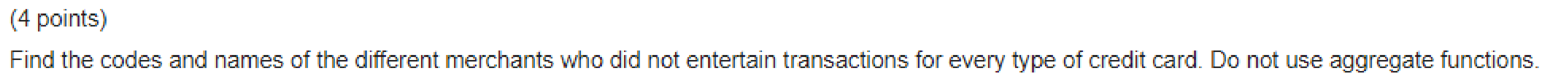
(4 points)
Find the social security number of the different customers who purchased something on Christmas day 2017 with their Visa credit card (the credit card type is "`visa`".)

## Query 8:



(4 points)
For each customer and for each credit card type, find how many credit cards of that type the customer owns. Print the customer's social security number, the credit card type and the number of credit cards of the given type owned. Print zero if a customer does not own a credit card of the given type.

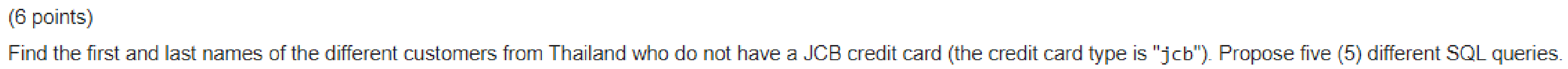
## Query 9:



(4 points)

Find the codes and names of the different merchants who did not entertain transactions for every type of credit card. Do not use aggregate functions.

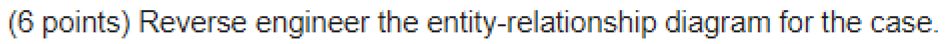
## Query 10:



(6 points)

Find the first and last names of the different customers from Thailand who do not have a JCB credit card (the credit card type is "`jcb`"). Propose five (5) different SQL queries.

## Entity relationship diagram



(6 points) Reverse engineer the entity-relationship diagram for the case.