Table A – “Customers”

customer_id	first_name	last_name	birthday	address	city	state	zip_code
10001	Amy	Smith	1963/4/1	100 S Glebe Rd	Arlington	VA	22204
10002	Boris	Hayward	1964/3/20	101 Ocean Ave	Los Angeles	CA	90402
10003	Charles	Gordon	1965/5/14	200 N Glebe Rd	Arlington	VA	22203
10004	Debby	Thompson	1966/10/3	201 Ocean Ave	Los Angeles	CA	90402
10005	Ella	Ryan	1967/1/1	300 N Glebe Rd	Arlington	VA	22203

Note: Identities are fictional

Table B – “Transactions”

transaction_id	customer_id	repeat_customer	amount
991	10001	Yes	$500.00
992	10001	Yes	$550.00
993	10002	No	$400.50
994	10003	No	$950.50
995	10004	Yes	$550.00

Table B2 – “Transactions”

transaction_id	customer_id	amount	payment_type
991	10001	$500.00	Credit Card
992	10001	$550.00	Credit Card
993	10002	$400.50	Credit Card
994	10003	$950.50	Credit Card
995	10004	$550.00	Credit Card

Table B3 – “Transactions”

transaction_id	customer_id	amount	payment_type
991	10001	$500.00	Debit Card
992	10001	$550.00	Cash
993	10002	$400.50	Cash
994	10003	$950.50	Credit Card
995	10004	$550.00	Debit Card

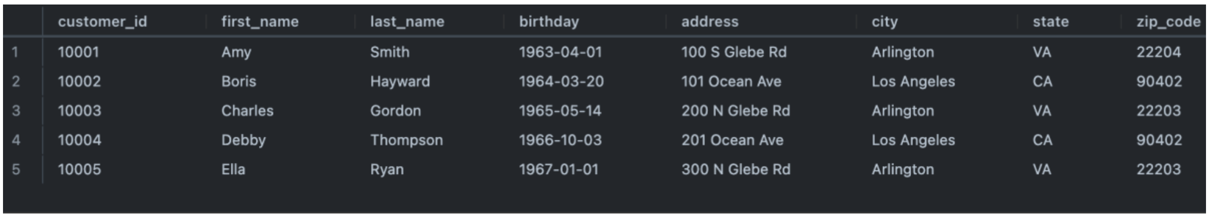
Key:

primary_key	foreign_key

1. Create Table A in SQL

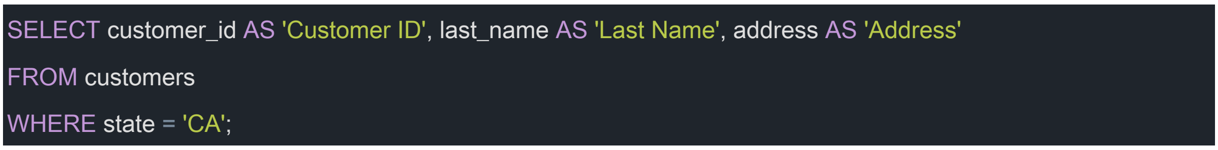
```
CREATE TABLE customers (
    customer_id INT AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR(25),
    last_name VARCHAR(25),
    birthday DATE,
    address VARCHAR(30),
    city VARCHAR(20),
    state VARCHAR(2),
    zip_code INT
);

/* Note: The use of AUTO_INCREMENT allows me to only enter the customer_id once,
after which customer_id automatically increases by 1 for each entry, as desired. */

INSERT INTO customers
VALUES(10001, 'Amy', 'Smith', '1963-04-01', '100 S Glebe Rd', 'Arlington', 'VA', 22204);
INSERT INTO customers(first_name, last_name, birthday, address, city, state, zip_code)
VALUES('Boris', 'Hayward', '1964-03-20', '101 Ocean Ave', 'Los Angeles', 'CA', 90402);
INSERT INTO customers(first_name, last_name, birthday, address, city, state, zip_code)
VALUES('Charles', 'Gordon', '1965-05-14', '200 N Glebe Rd', 'Arlington', 'VA', 22203);
INSERT INTO customers(first_name, last_name, birthday, address, city, state, zip_code)
VALUES('Debby', 'Thompson', '1966-10-03', '201 Ocean Ave', 'Los Angeles', 'CA', 90402);
INSERT INTO customers(first_name, last_name, birthday, address, city, state, zip_code)
VALUES('Ella', 'Ryan', '1967-01-01', '300 N Glebe Rd', 'Arlington', 'VA', 22203);

SELECT * FROM customers; --To return all of Table A as output
```

Result:



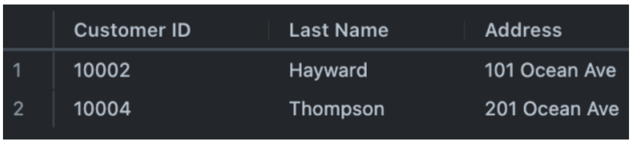
	customer_id	first_name	last_name	birthday	address	city	state	zip_code
1	10001	Amy	Smith	1963-04-01	100 S Glebe Rd	Arlington	VA	22204
2	10002	Boris	Hayward	1964-03-20	101 Ocean Ave	Los Angeles	CA	90402
3	10003	Charles	Gordon	1965-05-14	200 N Glebe Rd	Arlington	VA	22203
4	10004	Debby	Thompson	1966-10-03	201 Ocean Ave	Los Angeles	CA	90402
5	10005	Ella	Ryan	1967-01-01	300 N Glebe Rd	Arlington	VA	22203

2. Design a SQL query to pull customers living in the state of California (CA). *Output should include Customer ID, Last Name and Address.*



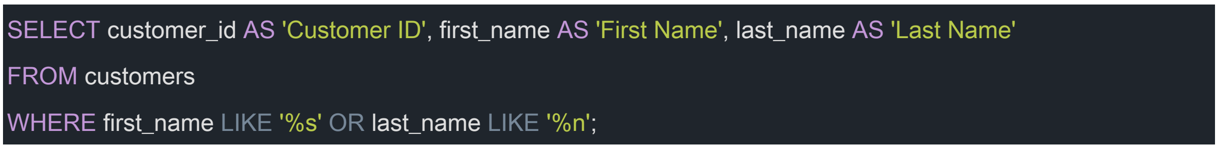
```
SELECT customer_id AS 'Customer ID', last_name AS 'Last Name', address AS 'Address'
FROM customers
WHERE state = 'CA';
```

Result:



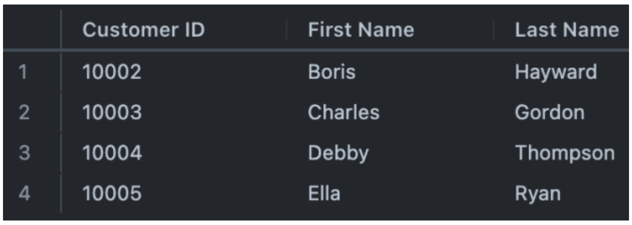
	Customer ID	Last Name	Address
1	10002	Hayward	101 Ocean Ave
2	10004	Thompson	201 Ocean Ave

3. Design a SQL query to pull customers whose first name ends in the letter “s” or last name ends in the letter “n”. *Output should include Customer ID, First Name and Last Name.*



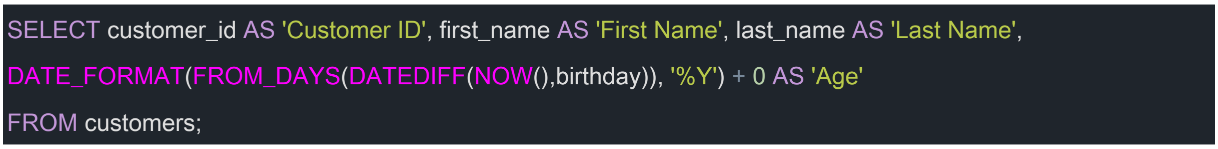
```
SELECT customer_id AS 'Customer ID', first_name AS 'First Name', last_name AS 'Last Name'
FROM customers
WHERE first_name LIKE '%s' OR last_name LIKE '%n';
```

Result:



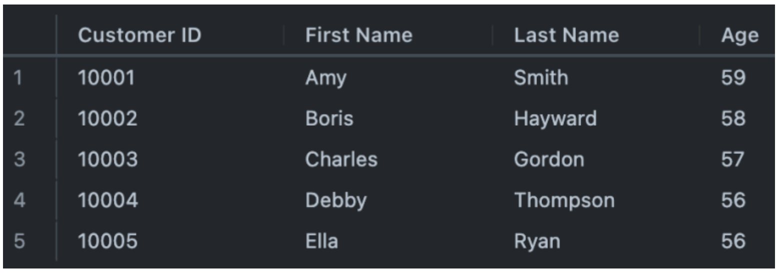
	Customer ID	First Name	Last Name
1	10002	Boris	Hayward
2	10003	Charles	Gordon
3	10004	Debby	Thompson
4	10005	Ella	Ryan

4. Design a SQL query to extract age of all customers using their birthdays. *Output should include Customer ID, First Name, Last Name and Age.*



```
SELECT customer_id AS 'Customer ID', first_name AS 'First Name', last_name AS 'Last Name',
DATE_FORMAT(FROM_DAYS(DATEDIFF(NOW(),birthday)), '%Y') + 0 AS 'Age'
FROM customers;
```

Result:

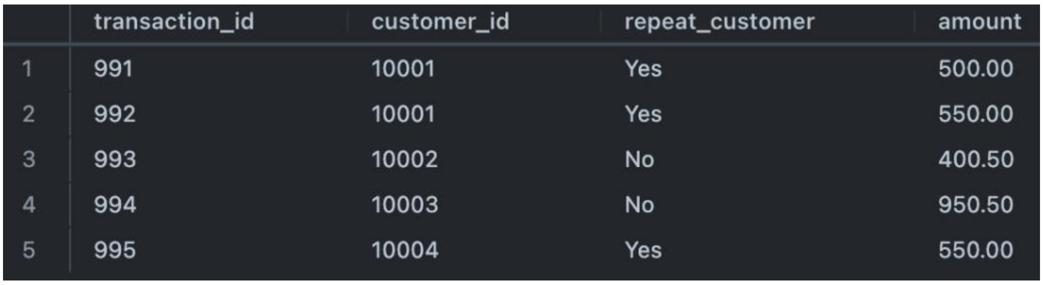


	Customer ID	First Name	Last Name	Age
1	10001	Amy	Smith	59
2	10002	Boris	Hayward	58
3	10003	Charles	Gordon	57
4	10004	Debby	Thompson	56
5	10005	Ella	Ryan	56

5. Create Table B in SQL

```
CREATE TABLE transactions (
    transaction_id INT AUTO_INCREMENT PRIMARY KEY,
    customer_id INT,
    repeat_customer VARCHAR(3),
    amount DEC(5,2),
    FOREIGN KEY(customer_id) REFERENCES customers(customer_id) ON DELETE SET NULL
);

/* Note: The use of ON DELETE SET NULL (rather than an alternative, such as ON
DELETE CASCADE) as we would like to keep the entry in the transaction table
(unlike ON DELETE CASCADE, which would delete the entry as a record of a purchase,
even if the entry in the customer table with the corresponding Customer ID is
deleted for whatever reason. */

INSERT INTO transactions (transaction_id, customer_id, repeat_customer, amount)
VALUES(991, 10001, 'Yes', 500.00);
INSERT INTO transactions (customer_id, repeat_customer, amount)
VALUES(10001, 'Yes', 550.00);
INSERT INTO transactions (customer_id, repeat_customer, amount)
VALUES(10002, 'No', 400.50);
INSERT INTO transactions (customer_id, repeat_customer, amount)
VALUES(10003, 'No', 950.50);
INSERT INTO transactions (customer_id, repeat_customer, amount)
VALUES(10004, 'Yes', 550.00);

SELECT * FROM transactions; --To return all of Table B as output
```

Result:

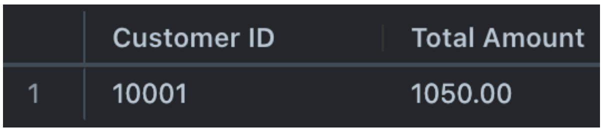


	transaction_id	customer_id	repeat_customer	amount
1	991	10001	Yes	500.00
2	992	10001	Yes	550.00
3	993	10002	No	400.50
4	994	10003	No	950.50
5	995	10004	Yes	550.00

6. Using Table A and Table B, design a SQL query to calculate total spending by Virgina (VA) residents born before 1965. *Output should include Customer ID (for the relevant customer) and Total Amount.*

```
SELECT customer_id AS 'Customer ID', SUM(amount) AS 'Total Amount'
FROM transactions
WHERE customer_id = (
  SELECT customer_id
  FROM customers
  WHERE state = 'VA' AND birthday < '1965-01-01')
GROUP BY customer_id;
```

Result:

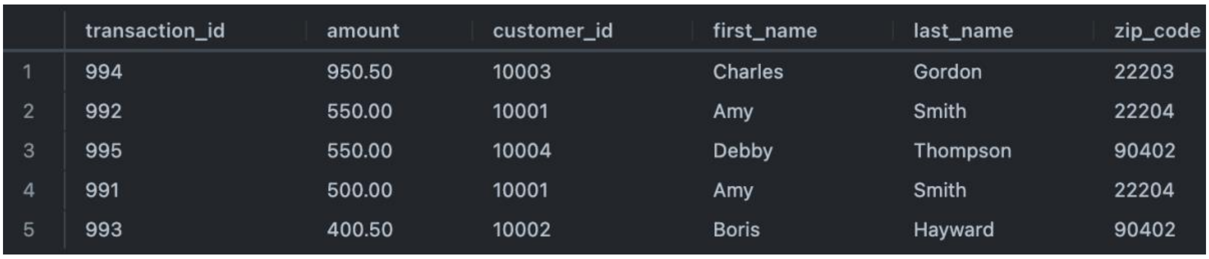


	Customer ID	Total Amount
1	10001	1050.00

7. Using Table A and Table B, design a SQL query to present, for each transaction in Table B, the corresponding customer information (ID, first name, last name and zip code) from Table A, sorted from biggest purchase to smallest. *Output should include Customer ID, First Name and Last Name, Zip Code, Transaction ID and Amount.*

```
SELECT
transactions.transaction_id, transactions.amount,
customers.customer_id, customers.first_name, customers.last_name, customers.zip_code
FROM customers JOIN transactions
ON customers.customer_id = transactions.customer_id
ORDER BY transactions.amount DESC;

/* Note: I could have also used RIGHT JOIN above in the place of JOIN to return an
identical output, since of the Customer IDs in the transactions table appear in the
customers table. */
```

Result:

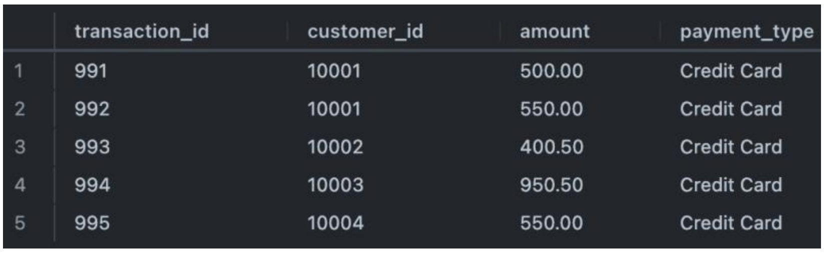


	transaction_id	amount	customer_id	first_name	last_name	zip_code
1	994	950.50	10003	Charles	Gordon	22203
2	992	550.00	10001	Amy	Smith	22204
3	995	550.00	10004	Debby	Thompson	90402
4	991	500.00	10001	Amy	Smith	22204
5	993	400.50	10002	Boris	Hayward	90402

8. The company wants to begin tracking payment type, so add this as a column to Table B and assume that the payment type is credit card for all transactions. Also – despite advice from colleagues – the new CEO wants to stop tracking whether transactions are from a repeat customer or not, so remove this data from the table. (i.e. Alter to Table B2)

```
ALTER TABLE transactions ADD
payment_type VARCHAR(15) DEFAULT 'Credit Card';
ALTER TABLE transactions DROP repeat_customer;

SELECT * FROM transactions; --To return all of Table B2 as output
```

Result:

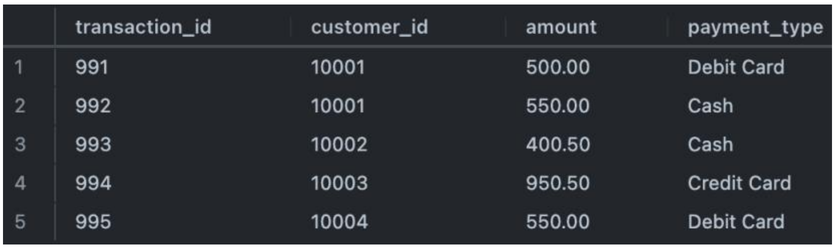


	transaction_id	customer_id	amount	payment_type
1	991	10001	500.00	Credit Card
2	992	10001	550.00	Credit Card
3	993	10002	400.50	Credit Card
4	994	10003	950.50	Credit Card
5	995	10004	550.00	Credit Card

9. The company finds that in some cases cash or debit cards were used, so update according to Table B3.

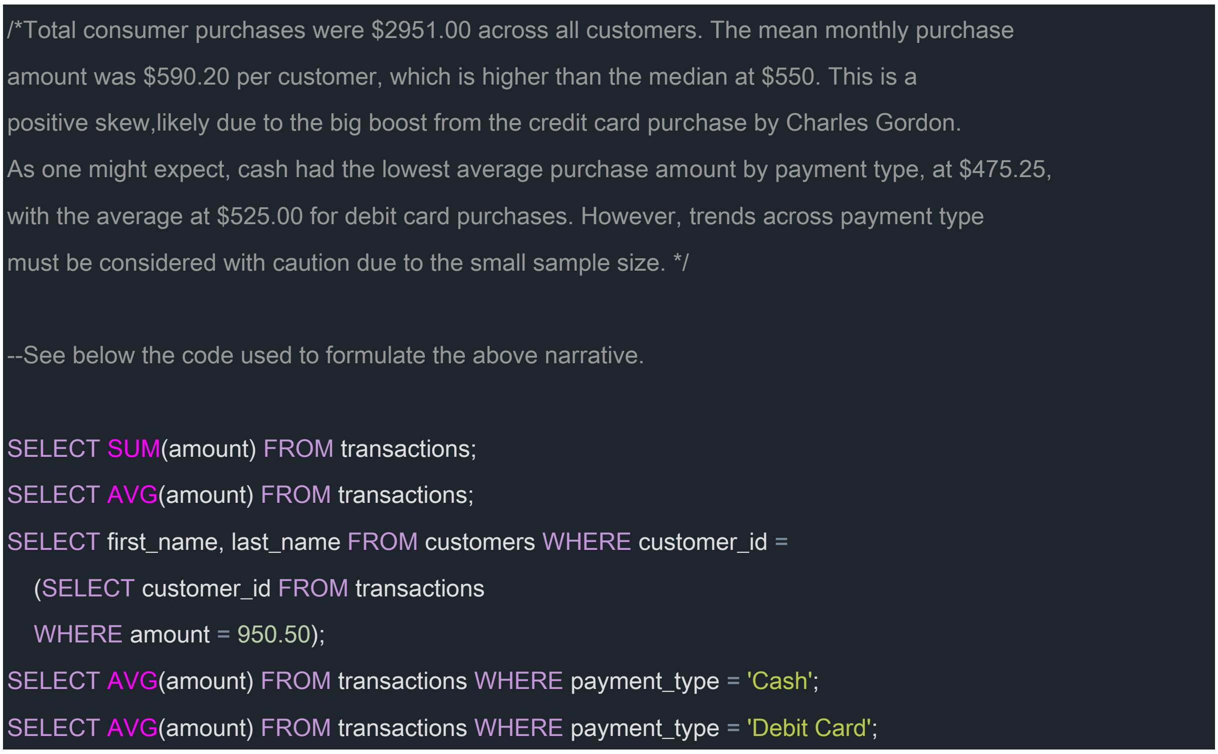
```
UPDATE transactions
SET payment_type = 'Debit Card'
WHERE transaction_id IN (991,995);
UPDATE transactions
SET payment_type = 'Cash'
WHERE transaction_id IN (992,993);

SELECT * FROM transactions; --To return all of Table B2 as output
```

Result:



	transaction_id	customer_id	amount	payment_type
1	991	10001	500.00	Debit Card
2	992	10001	550.00	Cash
3	993	10002	400.50	Cash
4	994	10003	950.50	Credit Card
5	995	10004	550.00	Debit Card

10. Construct a narrative around consumer purchases as shown in Table B3, considering the payment type. (5 sentences max)



```
/*Total consumer purchases were $2951.00 across all customers. The mean monthly purchase
amount was $590.20 per customer, which is higher than the median at $550. This is a
positive skew,likely due to the big boost from the credit card purchase by Charles Gordon.
As one might expect, cash had the lowest average purchase amount by payment type, at $475.25,
with the average at $525.00 for debit card purchases. However, trends across payment type
must be considered with caution due to the small sample size. */

--See below the code used to formulate the above narrative.

SELECT SUM(amount) FROM transactions;
SELECT AVG(amount) FROM transactions;
SELECT first_name, last_name FROM customers WHERE customer_id =
   (SELECT customer_id FROM transactions
   WHERE amount = 950.50);
SELECT AVG(amount) FROM transactions WHERE payment_type = 'Cash';
SELECT AVG(amount) FROM transactions WHERE payment_type = 'Debit Card';
```