

EASY

1.

```
nina — psql ◀ runpsql.sh — 80x24
SET
SET
SET
SET
SET
SET
SET
CREATE TABLE
ALTER TABLE
CREATE TABLE
ALTER TABLE
COPY 5191
COPY 99457
[sales=# \c sales
You are now connected to database "sales" as user "postgres".
sales=# -- How many unique customers are in the dataset?
sales=# SELECT COUNT(DISTINCT id) AS num_unique_customers
sales-# FROM customer;
      num_unique_customers
-----
                    5191
(1 row)

[sales=#
sales=# █
```

2.

```
nina — psql ◀ runpsql.sh — 80x24
sales=# -- 2. What are the different categories of products available? How many
unique categories are there?
sales=# SELECT DISTINCT category
sales-# FROM invoice;
      category
-----
Food & Beverage
Souvenir
Cosmetics
Books
Clothing
Toys
Shoes
Technology
(8 rows)

sales=#
sales=# SELECT COUNT(DISTINCT category) AS num_unique_categories
sales-# FROM invoice;
      num_unique_categories
-----
                        8
(1 row)
```

3.

```
nina — psql ◀ runpsql.sh — 80x24
(8 rows)

sales=#
sales=# SELECT COUNT(DISTINCT category) AS num_unique_categories
sales=# FROM invoice;
num_unique_categories
-----
                        8
(1 row)

sales=#
sales=# -- 3. Which payment method is the most popular? How many times was it used?
sales=# SELECT payment_method, COUNT(*) AS num_times_used
sales=# FROM invoice
sales=# GROUP BY payment_method
sales=# ORDER BY num_times_used DESC
sales=# LIMIT 1;
 payment_method | num_times_used
-----+-----
Cash            |          44447
(1 row)

sales=#
```

MEDIUM

1.

```
nina — psql ◀ runpsql.sh — 80x24

sales=# SELECT payment_method, COUNT(*) AS num_times_used
sales=# FROM invoice
sales=# GROUP BY payment_method
sales=# ORDER BY num_times_used DESC
sales=# LIMIT 1;
 payment_method | num_times_used
-----+-----
Cash            |          44447
(1 row)

sales=# -- 1. What are the three most popular categories, by total sales?
sales=# SELECT category, SUM(quantity * price) AS total_sales
sales=# FROM invoice
sales=# GROUP BY category
sales=# ORDER BY total_sales DESC
sales=# LIMIT 3;
 category | total_sales
-----+-----
Clothing  | 113996791.03997
Shoes     | 66553451.47000195
Technology | 57862350
(3 rows)

sales=#
```

2.

```
nina — psql • runpsql.sh — 80x24
sales=# SELECT category, SUM(quantity * price) AS total_sales
sales=# FROM invoice
sales=# GROUP BY category
sales=# ORDER BY total_sales DESC
sales=# LIMIT 3;
  category |      total_sales
-----+-----
Clothing   | 113996791.03997
Shoes      | 66553451.47000195
Technology |      57862350
(3 rows)

sales=# -- 2. What are the total sales attributed to customers over the age of 4
5?
sales=# SELECT SUM(i.quantity * i.price) AS total_sales_over_45
sales=# FROM invoice i
sales=# JOIN customer c ON i.customer_id = c.id
sales=# WHERE c.age > 45;
      total_sales_over_45
-----
      82039768.14999609
(1 row)

sales=#
```

HARD

1.

```
sales=# -- 1. Create a table similar to a pivot table showing the breakdown of sales across categories and decade age ranges
sales=# SELECT
sales=#     category,
sales=#     SUM(CASE WHEN age >= 10 AND age <= 19 THEN quantity * price ELSE 0 END) AS "10-19",
sales=#     SUM(CASE WHEN age >= 20 AND age <= 29 THEN quantity * price ELSE 0 END) AS "20-29",
sales=#     SUM(CASE WHEN age >= 30 AND age <= 39 THEN quantity * price ELSE 0 END) AS "30-39",
sales=#     SUM(CASE WHEN age >= 40 AND age <= 49 THEN quantity * price ELSE 0 END) AS "40-49",
sales=#     SUM(CASE WHEN age >= 50 THEN quantity * price ELSE 0 END) AS "50+"
sales=# FROM
sales=#     invoice i
sales=# JOIN
sales=#     customer c ON i.customer_id = c.id
sales=# GROUP BY
sales=#     category;
  category |      10-19 |      20-29 |      30-39 |      40-49 |      50+
-----+-----+-----+-----+-----+-----
Books      | 12801.75   | 79461.7500000003 | 84143.0999999998 | 87203.4   | 236930.85000000015
Clothing   | 1927413.8399999961 | 10250732.80000003 | 11354126.960000057 | 11215189.920000058 | 33245563.120000377
Cosmetics  | 109619.36000000002 | 605915.3200000016 | 723544.7000000014 | 665034.9600000016 | 1911385.9400000158
Food & Beverage | 13582.30999999994 | 73491.95999999973 | 87246.85999999983 | 89553.2899999998 | 239785.03999999852
Shoes      | 993281.349999996 | 5924878.24   | 7200839.660000004 | 7025590.020000003 | 18940765.029999897
Souvenir   | 12375.15   | 54943.3199999999 | 64092.71999999985 | 66391.7999999999 | 172360.61999999976
Technology | 1129800    | 5261550     | 6159300     | 5954550     | 15491700
Toys       | 60426.24000000002 | 385100.7999999998 | 420582.3999999982 | 390906.8799999984 | 1136199.679999999
(8 rows)
```