```
-- Restaurant Owners
-- 5 Tables
-- 1x Fact, 4x Dimension
-- search google, how to add foreign key
-- write SQL 3-5 queries analyze data
-- 1x subquery/ with

-- 1st dimension table: membership card id
CREATE TABLE membership (
  member_id VARCHAR PRIMARY KEY,
  firstname VARCHAR,
  lastname VARCHAR,
  date_of_birth DATE,
  date_of_apply DATE
);

INSERT INTO membership VALUES
  ('0001', 'Matt', 'Reeves', '1980-12-11', '2022-01-10'),
  ('0002', 'Jeremy', 'Iron', '1999-08-09', '2022-02-13'),
  ('0003', 'Von', 'Weera', '2004-03-28', '2022-05-27');

-- 2nd dimension table: Order Channel
CREATE TABLE order_channel (
  channel_id INT PRIMARY KEY,
  channel_name VARCHAR
);

INSERT INTO order_channel VALUES
  (1, 'Restaurant'),
  (2, 'Grab'),
  (3, 'Lineman'),
  (4, 'Robinhood');

-- 3rd dimension table: Order Type
CREATE TABLE order_type (
  type_id INT PRIMARY KEY,
  type_name VARCHAR
);

INSERT INTO order_type VALUES
  (1, 'For here'),
  (2, 'Take away'),
  (3, 'Self-pick up'),
  (4, 'Delivery');

-- 4th dimension table: Promotion Type
CREATE TABLE promotion (
  pro_id INT PRIMARY KEY,
```

```
  pro_name VARCHAR
);

INSERT INTO promotion VALUES
  (1, 'Own restaurant'),
  (2, 'Delivery app');

-- Fact Table: Orders
CREATE TABLE orders (
  order_id INT PRIMARY KEY,
  order_date DATE,
  amount REAL,
  member_id VARCHAR,
  order_channel INT,
  order_type INT,
  promotion INT,
  FOREIGN KEY(member_id) REFERENCES membership(member_id),
  FOREIGN KEY(order_channel) REFERENCES order_channel(channel_id),
  FOREIGN KEY(order_type) REFERENCES order_type(type_id),
  FOREIGN KEY(promotion) REFERENCES promotion(pro_id)
);

INSERT INTO orders VALUES
  (0001, '2022-03-24', 250, null, 1, 1, null),
  (0002, '2022-03-24', 123, '0002', 1, 2, 1),
  (0003, '2022-03-28', 112, null, 2, 3, null),
  (0004, '2022-04-01', 380, '0001', 1, 1, 1),
  (0005, '2022-04-11', 400, '0002', 4, 4, 2),
  (0006, '2022-04-27', 65, null, 1, 1, null),
  (0007, '2022-05-04', 1280, '0003', 3, 3, 2),
  (0008, '2022-06-18', 769, null, 2, 3, 2),
  (0009, '2022-06-23', 134, null, 2, 4, null),
  (0010, '2022-06-23', 100, null, 1, 1, 1);

.mode markdown
.header on

-- channel
  SELECT
  chan.channel_name,
  COUNT(ord.order_channel)
FROM orders AS ord
JOIN order_channel AS chan on ord.order_channel = chan.channel_id
GROUP BY chan.channel_name
ORDER BY 2 DESC;
--

--             promotion
SELECT
  pro.pro_name,
```

```
  COUNT(ord.promotion)
FROM orders AS ord JOIN promotion AS pro
ON ord.promotion = pro.pro_id
WHERE ord.promotion = 1;
--           3

-- membership       promotion     channel
WITH member_pro AS(
  SELECT * FROM orders
  WHERE promotion IS NOT NULL AND member_id IS NOT NULL
)

SELECT
  pro.pro_name,
  COUNT(mpro.promotion)
FROM member_pro AS mpro
JOIN promotion AS pro ON mpro.promotion = pro.pro_id
GROUP BY pro.pro_name;
--                      2

--      pro
WITH member_pro AS(
  SELECT * FROM orders
  WHERE promotion IS NOT NULL AND member_id IS NOT NULL
)

SELECT
  ord.channel_name,
  COUNT(mpro.promotion)
FROM member_pro AS mpro
JOIN order_channel AS ord ON mpro.order_channel = ord.channel_id
WHERE ord.channel_name <> 'Restaurant'
GROUP BY ord.channel_name;
--             Lineman    Robinhood

--
SELECT
  strftime('%m', order_date) AS montham,
  SUM(amount)
FROM orders
GROUP BY montham
ORDER BY 2 DESC;
--             5

--                member
WITH n_memorder AS(
  SELECT
  order_id,
  CASE
    WHEN member_id IS NULL THEN 'Non member'
```

```
    ELSE 'Membership'
  END AS segment
FROM orders)

SELECT
  segment,
  COUNT(segment)
FROM n_memorder
GROUP BY segment
ORDER BY 2 DESC;
--          4           member
```