Deliverable 2 (Group 62)

> Members: Weiming Guo, Hengxian Jiang, Yingjie Xu, Helen Ren

Q2. table creation statement

customer

- create

```
CREATE TABLE customer
(
   phone_number VARCHAR(15) NOT NULL
  ,canme VARCHAR(40)
  ,address VARCHAR(100)
  ,PRIMARY KEY(phone_number)
);
```

- description

```
Table "cs421g62.orders"
    Column    |         Type          | Modifiers
--------------+-----------------------+-----------
 order_number | character varying(30) | not null
 tips         | double precision      |
Indexes:
    "orders_pkey" PRIMARY KEY, btree (order_number)
Referenced by:
    TABLE "contain" CONSTRAINT "contain_order_number_fkey" FOREIGN KEY
(order_number) REFERENCES orders(order_number)
    TABLE "dine_in_orders" CONSTRAINT "dine_in_orders_order_number_fkey"
FOREIGN KEY (order_number) REFERENCES orders(order_number)
    TABLE "served_by" CONSTRAINT "served_by_order_number_fkey" FOREIGN KEY
(order_number) REFERENCES orders(order_number)
```

staff

- create

```
CREATE TABLE staff
(
   sid INTEGER NOT NULL
  ,sname VARCHAR(40)
  ,working_schdule VARCHAR(12)
  ,salary FLOAT8
  ,PRIMARY KEY(sid)
);
```

- description

```
\d staff
               Table "cs421g62.staff"
     Column      |         Type          | Modifiers
-----------------+-----------------------+-----------
```

```
 sid              | integer               | not null
 sname            | character varying(12) |
 working_schdule | character varying(12) |
 salary           | double precision      |
Indexes:
    "staff_pkey" PRIMARY KEY, btree (sid)
Referenced by:
    TABLE "chef" CONSTRAINT "chef_sid_fkey" FOREIGN KEY (sid) REFERENCES
staff(sid)
    TABLE "delivery_guy" CONSTRAINT "delivery_guy_sid_fkey" FOREIGN KEY
(sid) REFERENCES staff(sid)
    TABLE "delivery_orders" CONSTRAINT "delivery_orders_sid_fkey" FOREIGN
KEY (sid) REFERENCES staff(sid)
    TABLE "served_by" CONSTRAINT "served_by_sid_fkey" FOREIGN KEY (sid)
REFERENCES staff(sid)
    TABLE "waiter" CONSTRAINT "waiter_sid_fkey" FOREIGN KEY (sid) REFERENCES
staff(sid)
```

platform

- create

```
CREATE TABLE platform
(
   pname VARCHAR(30) NOT NULL
  ,url VARCHAR(100)
  ,PRIMARY KEY(pname)
);
```

- description

```
\d platform
          Table "cs421g62.platform"
 Column |          Type          | Modifiers
--------+------------------------+-----------
 pname  | character varying(30)  | not null
 url    | character varying(100) |
Indexes:
    "platform_pkey" PRIMARY KEY, btree (pname)
Referenced by:
    TABLE "delivery_orders" CONSTRAINT "delivery_orders_pname_fkey" FOREIGN
KEY (pname) REFERENCES platform(pname)
```

orders

- create

```
CREATE TABLE orders
(
   order_number VARCHAR(30) NOT NULL
  ,tips FLOAT8
  ,PRIMARY KEY(order_number)
);
```

- description

```
\d orders
             Table "cs421g62.orders"
    Column    |         Type          | Modifiers
--------------+-----------------------+-----------
 order_number | character varying(30) | not null
 tips         | double precision      |
Indexes:
    "orders_pkey" PRIMARY KEY, btree (order_number)
Referenced by:
    TABLE "contain" CONSTRAINT "contain_order_number_fkey" FOREIGN KEY
(order_number) REFERENCES orders(order_number)
    TABLE "dine_in_orders" CONSTRAINT "dine_in_orders_order_number_fkey"
FOREIGN KEY (order_number) REFERENCES orders(order_number)
    TABLE "served_by" CONSTRAINT "served_by_order_number_fkey" FOREIGN KEY
(order_number) REFERENCES orders(order_number)
```

dine_in_orders

- create

```
CREATE TABLE dine_in_orders
(
   order_number VARCHAR(30) NOT NULL
  ,phone_number VARCHAR(15) NOT NULL
  ,PRIMARY KEY(order_number)
  ,FOREIGN KEY(order_number) REFERENCES orders(order_number)
  ,FOREIGN KEY(phone_number) REFERENCES customer(phone_number)
);
```

- description

```
\d dine_in_orders
         Table "cs421g62.dine_in_orders"
    Column    |         Type          | Modifiers
--------------+-----------------------+-----------
 order_number | character varying(30) | not null
 phone_number | character varying(15) | not null
Indexes:
    "dine_in_orders_pkey" PRIMARY KEY, btree (order_number)
Foreign-key constraints:
    "dine_in_orders_order_number_fkey" FOREIGN KEY (order_number) REFERENCES
orders(order_number)
    "dine_in_orders_phone_number_fkey" FOREIGN KEY (phone_number) REFERENCES
customer(phone_number)
```

delivery_orders

- create

```
CREATE TABLE delivery_orders
(
   order_number VARCHAR(30) NOT NULL
  ,phone_number VARCHAR(15) NOT NULL
  ,sid INTEGER NOT NULL
  ,pname VARCHAR(30)
  ,delivery_fee FLOAT8
  ,PRIMARY KEY(order_number)
  ,FOREIGN KEY(order_number) REFERENCES orders(order_number)
  ,FOREIGN KEY(sid) REFERENCES staff(sid)
  ,FOREIGN KEY(pname) REFERENCES platform(pname)
);
```

- description

```
\d delivery_orders
         Table "cs421g62.delivery_orders"
    Column    |         Type          | Modifiers
--------------+-----------------------+-----------
 order_number | character varying(30) | not null
 phone_number | character varying(15) | not null
 sid          | integer               | not null
 pname        | character varying(30) |
 delivery_fee | double precision      |
Indexes:
    "delivery_orders_pkey" PRIMARY KEY, btree (order_number)
Foreign-key constraints:
    "delivery_orders_pname_fkey" FOREIGN KEY (pname) REFERENCES
platform(pname)
    "delivery_orders_sid_fkey" FOREIGN KEY (sid) REFERENCES staff(sid)
```

waiter

- create

```
CREATE TABLE waiter
(
   sid INTEGER NOT NULL
  ,FOREIGN KEY(sid) REFERENCES staff(sid)
  ,PRIMARY KEY(sid)
);
```

- description

```
\d waiter
   Table "cs421g62.waiter"
 Column |  Type   | Modifiers
--------+---------+-----------
 sid    | integer | not null
Indexes:
    "waiter_pkey" PRIMARY KEY, btree (sid)
Foreign-key constraints:
    "waiter_sid_fkey" FOREIGN KEY (sid) REFERENCES staff(sid)
```

delivery_guy

- create

```
CREATE TABLE delivery_guy
(
   delivery_method VARCHAR(12)
  ,phone_number VARCHAR(15)
  ,sid INTEGER NOT NULL
  ,FOREIGN KEY(sid) REFERENCES staff(sid)
  ,PRIMARY KEY(sid)
);
```

- description

```
\d delivery_guy
            Table "cs421g62.delivery_guy"
     Column      |         Type          | Modifiers
-----------------+-----------------------+-----------
 delivery_method | character varying(12) |
 phone_number    | character varying(15) |
 sid             | integer               | not null
Indexes:
    "delivery_guy_pkey" PRIMARY KEY, btree (sid)
Foreign-key constraints:
    "delivery_guy_sid_fkey" FOREIGN KEY (sid) REFERENCES staff(sid)
```

chef

- create

```
CREATE TABLE chef
(
   proficiency INTEGER
  ,cooking_style VARCHAR(12)
  ,sid INTEGER NOT NULL
  ,FOREIGN KEY(sid) REFERENCES staff(sid)
  ,PRIMARY KEY(sid)
);
```

- description

```
\d chef
               Table "cs421g62.chef"
    Column     |         Type          | Modifiers
---------------+-----------------------+-----------
 proficiency   | character varying(12) |
 cooking_style | character varying(12) |
 sid           | integer               | not null
Indexes:
    "chef_pkey" PRIMARY KEY, btree (sid)
Foreign-key constraints:
    "chef_sid_fkey" FOREIGN KEY (sid) REFERENCES staff(sid)
Referenced by:
    TABLE "cooked_by" CONSTRAINT "cooked_by_sid_fkey" FOREIGN KEY (sid)
REFERENCES chef(sid)
```

dish

- create

```
CREATE TABLE dish
(
   dish_name VARCHAR(50) NOT NULL
  ,price FLOAT8
  ,PRIMARY KEY(dish_name)
);
```

- description

```
\d dish
             Table "cs421g62.dish"
  Column   |         Type          | Modifiers
-----------+-----------------------+-----------
 dish_name | character varying(30) | not null
 price     | double precision      |
Indexes:
    "dish_pkey" PRIMARY KEY, btree (dish_name)
Referenced by:
    TABLE "contain" CONSTRAINT "contain_dish_name_fkey" FOREIGN KEY
(dish_name) REFERENCES dish(dish_name)
    TABLE "cooked_by" CONSTRAINT "cooked_by_dish_name_fkey" FOREIGN KEY
(dish_name) REFERENCES dish(dish_name)
```

reservation

- create

```
CREATE TABLE reservation
(
   rdate DATE
  ,phone_number VARCHAR(15)
  ,timeslot TIME
  ,PRIMARY KEY(rdate, phone_number)
  ,FOREIGN KEY(phone_number) REFERENCES customer(phone_number)
);
```

- description

```
\d reservation
           Table "cs421g62.reservation"
    Column    |          Type          | Modifiers
--------------+------------------------+-----------
 rdate        | date                   | not null
 phone_number | character varying(15)  | not null
 timeslot     | time with time zone    |
Indexes:
    "reservation_pkey" PRIMARY KEY, btree (rdate, phone_number)
Foreign-key constraints:
    "reservation_phone_number_fkey" FOREIGN KEY (phone_number) REFERENCES
customer(phone_number)
```

contain

- create

```
CREATE TABLE contain
(
   order_number VARCHAR(20)
  ,dish_name VARCHAR(50)
  ,quantity INTEGER
  ,PRIMARY KEY(order_number, dish_name)
  ,FOREIGN KEY(order_number) REFERENCES orders(order_number)
  ,FOREIGN KEY(dish_name) REFERENCES dish(dish_name)
);
```

- description

```
\d contain
             Table "cs421g62.contain"
    Column    |          Type          | Modifiers
--------------+------------------------+-----------
 order_number | character varying(20)  | not null
 dish_name    | character varying(30)  | not null
 quantity     | integer                |
Indexes:
    "contain_pkey" PRIMARY KEY, btree (order_number, dish_name)
Foreign-key constraints:
    "contain_dish_name_fkey" FOREIGN KEY (dish_name) REFERENCES
dish(dish_name)
    "contain_order_number_f
```

cooked_by

- create

```
CREATE TABLE cooked_by
(
   dish_name VARCHAR(50)
  ,sid INTEGER
  ,PRIMARY KEY(dish_name, sid)
  ,FOREIGN KEY(sid) REFERENCES chef(sid)
  ,FOREIGN KEY(dish_name) REFERENCES dish(dish_name)
);
```

- description

```
\d cooked_by
          Table "cs421g62.cooked_by"
  Column   |         Type          | Modifiers
-----------+-----------------------+-----------
 dish_name | character varying(30) | not null
 sid       | integer               | not null
Indexes:
    "cooked_by_pkey" PRIMARY KEY, btree (dish_name, sid)
Foreign-key constraints:
    "cooked_by_dish_name_fkey" FOREIGN KEY (dish_name) REFERENCES
dish(dish_name)
    "cooked_by_sid_fkey" FOREIGN KEY (sid) REFERENCES chef(sid)
```

served_by

- create

```
CREATE TABLE served_by
(
   order_number VARCHAR(20)
  ,sid INTEGER
  ,PRIMARY KEY(order_number, sid)
  ,FOREIGN KEY(order_number) REFERENCES orders(order_number)
  ,FOREIGN KEY(sid) REFERENCES staff(sid)
);
```

- description

```
\d served_by
            Table "cs421g62.served_by"
    Column    |         Type          | Modifiers
--------------+-----------------------+-----------
 order_number | character varying(20) | not null
 sid          | integer               | not null
Indexes:
    "served_by_pkey" PRIMARY KEY, btree (order_number, sid)
Foreign-key constraints:
    "served_by_order_number_fkey" FOREIGN KEY (order_number) REFERENCES
orders(order_number)
    "served_by_sid_fkey" FOREIGN KEY (sid) REFERENCES staff(sid)
```

Q3. 5 example `INSERT` commands

- Insertions:

```
cs421=> INSERT INTO reservation VALUES( '2020-01-29','514-123-8900','19:30');
INSERT 0 1
cs421=> INSERT INTO reservation VALUES( '2020-01-30','514-421-1768','16:30');
INSERT 0 1
cs421=> INSERT INTO reservation VALUES( '2020-01-21','514-141-5854','17:25');
INSERT 0 1
cs421=> INSERT INTO reservation VALUES( '2020-01-22','514-133-1567','16:45');
INSERT 0 1
cs421=> INSERT INTO reservation VALUES( '2020-01-23','514-140-1264','16:10');
INSERT 0 1
```

- results:

```
cs421=> SELECT * FROM reservation;
   rdate    | phone_number | timeslot
------------+--------------+----------
 2020-01-29 | 514-123-8900 | 19:30:00
 2020-01-30 | 514-421-1768 | 16:30:00
 2020-01-21 | 514-141-5854 | 17:25:00
 2020-01-22 | 514-133-1567 | 16:45:00
 2020-01-23 | 514-140-1264 | 16:10:00
(5 rows)
```

Q4. Display table

staff

```
cs421=> SELECT * FROM staff LIMIT 5;
 sid |     sname      | working_schdule | salary
-----+----------------+-----------------+--------
   1 | Samuel Randall | morning         |     12
   2 | Shala Tang     | morning         |     12
   3 | Macie Finely   | afternoon       |     12
   4 | Jeffrey Frye   | afternoon       |     12
   5 | Anna Black     | evening         |     12
(5 rows)
```

customer

```
cs421=> SELECT * FROM customer LIMIT 5;
 phone_number |     canme      |       address
--------------+----------------+----------------------
 514-123-8900 | Nimrah Carter   | 190  Lynden Road
 514-421-1768 | Tiana House     | 1528  Halsey Avenue
 514-141-5854 | Armaan Gibbons | 3569  Lockhart Drive
 514-133-1567 | Jevan Hope      | 3020  rue Parc
 514-140-1264 | Nettie Cline    | 2318  Royal Avenue
(5 rows)
```

platform

```
cs421=> SELECT * FROM platform LIMIT 5;
      pname       |              url
------------------+--------------------------------
 Uber Eats        | https://www.ubereats.com/en-CA
 Skip The Dishes  | https://www.skipthedishes.com/
 Foodora          | https://www.foodora.ca/
 Food Highway     | https://www.foodhwy.com/
 Fantuan Delivery | https://en.fantuan.ca/
(5 rows)
```

orders

```
cs421=> SELECT * FROM orders LIMIT 5;
 order_number | tips
--------------+------
 001          |  2.1
 002          |  1.5
 003          |  1.2
 004          |    3
 005          |  2.3
(5 rows)
```

dine_in_orders

```
cs421=> SELECT * FROM dine_in_orders LIMIT 5;
 order_number | phone_number
--------------+--------------
 001          | 514-123-8900
 002          | 514-421-1768
 003          | 514-141-5854
 004          | 514-133-1567
 005          | 514-140-1264
(5 rows)
```

delivery_orders

```
cs421=> SELECT * FROM delivery_orders LIMIT 5;
 order_number | phone_number | sid |      pname      | delivery_fee
--------------+--------------+-----+-----------------+--------------
 006          | 514-202-1544 |   7 | Uber Eats       |            5
 007          | 514-303-2334 |   9 | Skip The Dishes |            3
 008          | 514-404-2154 |  10 | Foodora         |          2.5
 009          | 514-273-3594 |   9 | Food Highway    |          2.7
 014          | 514-273-3594 |  10 | Food Highway    |          2.7
(5 rows)
```

waiter

```
cs421=> SELECT * FROM waiter LIMIT 5;
 sid
-----
   1
   2
   3
   4
   5
(5 rows)
```

delivery_guy

```
cs421=> SELECT * FROM delivery_guy LIMIT 5;
 delivery_method | phone_number | sid
-----------------+--------------+-----
 Car             | 514-999-9999 |   7
 Bike            | 514-888-8888 |   8
 On foot         | 514-777-7777 |   9
 Car             | 514-666-6666 |  10
 Bike            | 514-555-5555 |  11
(5 rows)
```

chef

```
cs421=> SELECT * FROM chef LIMIT 5;
 proficiency | cooking_style | sid
-------------+---------------+-----
           2 | Cutting       |  13
           3 | Steaming      |  14
           4 | Braising      |  15
           4 | Braising      |  16
           5 | Stir-frying   |  17
(5 rows)
```

dish

```
cs421=> SELECT * FROM dish LIMIT 5;
            dish_name             | price
----------------------------------+-------
 Tenderized Truffles & Yak        | 50.99
 Sautéed Dark Beer Pork           |  28.9
 Tenderized Parmesan Lobster      |  99.9
 Simmered Peas & Mushroom Oysters |  66.6
 Breaded Cucumber & Lime Pizza    |  19.9
(5 rows)
```

reservation

```
cs421=> SELECT * FROM reservation LIMIT 5;
   rdate    | phone_number | timeslot
------------+--------------+----------
 2020-01-29 | 514-123-8900 | 19:30:00
 2020-01-30 | 514-421-1768 | 16:30:00
 2020-01-21 | 514-141-5854 | 17:25:00
 2020-01-22 | 514-133-1567 | 16:45:00
 2020-01-23 | 514-140-1264 | 16:10:00
(5 rows)
```

contain

```
cs421=> SELECT * FROM contain LIMIT 5;
 order_number |         dish_name         | quantity
--------------+---------------------------+----------
 001          | Tenderized Truffles & Yak |        2
 001          | Guoyou pork               |        1
 001          | General Tsos Chicken      |        2
 002          | boiled spicy fish         |        2
 002          | beef pho                  |        2
(5 rows)
```

cooked_by

```
cs421=> SELECT * FROM cooked_by LIMIT 5;
            dish_name             | sid
----------------------------------+-----
 Tenderized Truffles & Yak        |  13
 Sautéed Dark Beer Pork           |  14
 Tenderized Parmesan Lobster      |  15
 Simmered Peas & Mushroom Oysters |  16
 Breaded Cucumber & Lime Pizza    |  17
(5 rows)
```

served_by

```
cs421=> SELECT * FROM served_by LIMIT 5;
 order_number | sid
--------------+-----
 001          |   1
 002          |   2
 003          |   3
 004          |   4
 005          |   5
(5 rows)
```

Q5. Queries

Query 1

Description:

> List the `dish_name` of the `dish` which sold most, as well as the portion it was sold

SQL query:

```
SELECT dish_name ,COUNT(*) AS dcount FROM contain
GROUP BY dish_name
HAVING COUNT(*) = (
    SELECT MAX(y.dcount) FROM
    (SELECT dish_name ,COUNT(*) AS dcount FROM contain GROUP BY dish_name) y
);
```

Result:

```
          dish_name          | dcount
-----------------------------+--------
 Tenderized Parmesan Lobster |      4
 Tenderized Truffles & Yak   |      4
(2 rows)
```

Query 2

Description:

> Display information about all the `delivery orders`. The query will first show the `order number` and then display the `sid` and `name` of the `delivery guy`. It will also show the customer's information to the corresponding order number.

SQL query:

```
SELECT t2.order_number, s.sname AS delivery_guy_name, t2.sid AS
delivery_guy_sid,
    t2.delivery_guy_phone, t2.cus_name, t2.cus_phone, t2.cus_address
FROM staff s
INNER JOIN
```

```
    (
        SELECT t1.order_number, t1.cus_name, t1.cus_phone, t1.cus_address,
            guy.phone_number AS delivery_guy_phone, t1.sid AS sid
        FROM delivery_guy guy
        INNER JOIN
        (
            SELECT cus.phone_number AS cus_phone, cus.address AS cus_address,
                cus.canme AS cus_name, sid, order_number
            FROM customer cus INNER JOIN delivery_orders d
            ON cus.phone_number = d.phone_number
        ) t1
        ON t1.sid = guy.sid
    ) t2
ON s.sid = t2.sid;
```

Result:

```
 order_number | delivery_guy_name | delivery_guy_sid | delivery_guy_phone |
cus_name    |  cus_phone   |       cus_address
--------------+-------------------+------------------+--------------------+-----
--------+--------------+--------------------------
 006          | Annie Carr        |                7 | 514-999-9999       | Ava
Pollard | 514-202-1544 | 1120  th Avenue
 007          | Jodie Stuart      |                9 | 514-777-7777       |
Teegan Liu  | 514-303-2334 | 3417  Brew Creek Rd
 008          | Cassie Noble      |               10 | 514-666-6666       | Jing
Guo    | 514-404-2154 | 283  Jasper Avenue
 009          | Jodie Stuart      |                9 | 514-777-7777       | Rong
Huang  | 514-273-3594 | 3242  Boulevard Cremazie
 014          | Cassie Noble      |               10 | 514-666-6666       | Rong
Huang  | 514-273-3594 | 3242  Boulevard Cremazie
 015          | Cara Allenr       |                8 | 514-888-8888       |
Teegan Liu  | 514-303-2334 | 3417  Brew Creek Rd
(6 rows)
```

Query 3

Description:

> List all `reservations` ordered by reservation date, then by timeslot within each reservation date. The derived table will display the names and phone numbers of those who have made an reservation, along with the reservation date and timeslot

SQL query:

```
SELECT canme, cus.phone_number, rdate, timeslot
FROM customer AS cus, reservation AS res
WHERE cus.phone_number = res.phone_number
ORDER BY rdate, timeslot;
```

Result:

```
     canme      | phone_number |   rdate    | timeslot
----------------+--------------+------------+----------
 Ava Pollard    | 514-202-1544 | 2020-01-15 | 18:20:00
 Armaan Gibbons | 514-141-5854 | 2020-01-21 | 17:25:00
 Jing Guo       | 514-404-2154 | 2020-01-22 | 12:30:00
 Jevan Hope     | 514-133-1567 | 2020-01-22 | 16:45:00
 Rong Huang     | 514-273-3594 | 2020-01-23 | 11:45:00
 Nettie Cline   | 514-140-1264 | 2020-01-23 | 16:10:00
 Tiana House    | 514-421-1768 | 2020-01-25 | 12:20:00
 Nimrah Carter  | 514-123-8900 | 2020-01-29 | 19:30:00
 Tiana House    | 514-421-1768 | 2020-01-30 | 16:30:00
 Teegan Liu     | 514-303-2334 | 2020-02-19 | 18:10:00
(10 rows)
```

Query 4

Description:

> Select best `chef` for each of the `cooking style`

SQL query:

```
SELECT S.sname, t3.proficiency, t3.cooking_style,S.sid FROM staff S JOIN
(SELECT chef.sid, chef.proficiency,chef.cooking_style FROM chef
INNER JOIN
((SELECT t1.cooking_style, MAX(t1.proficiency) AS maxProficiency FROM
  (SELECT DISTINCT S.sname, S.sid, C.proficiency, C.cooking_style FROM staff S
  JOIN
  chef C ON C.sid = S.sid)t1
GROUP BY t1.cooking_style)) t2
ON chef.proficiency = t2.maxProficiency AND t2.cooking_style =
chef.cooking_style)t3
ON S.sid = t3.sid;
```

Result:

```
    sname     | proficiency | cooking_style
--------------+-------------+---------------
 Lacie Barr   |           2 | Cutting
 Isla Greer   |           3 | Steaming
 Meadow Moran |           4 | Braising
 Steve Marwan |           4 | Braising
 Melanie Hill |           5 | Stir-frying
(5 rows)
```

Query 5

Description:

> Total revenue for all `orders`

SQL query:

```
SELECT SUM(D.price) AS total_revenue FROM dish D
JOIN
  (SELECT * FROM orders O
  JOIN
  contain C ON O.order_number = C.order_number)t1
ON D.dish_name = t1.dish_name;
```

Result:

```
total_revenue
---------------
       1198.51
(1 row)
```

Q6. Data modification

1. promotion of the best chef according to the rating

- description

  > We would increase the salary of chef with the highest rating

- statement

  ```
  UPDATE staff
  SET salary = salary+1
  WHERE sid IN
  (SELECT S.sid FROM staff S JOIN
  (SELECT chef.sid, chef.proficiency,chef.cooking_style FROM chef
  INNER JOIN
  ((SELECT t1.cooking_style, MAX(t1.proficiency) AS maxProficiency FROM
    (SELECT DISTINCT S.sname, S.sid, C.proficiency, C.cooking_style FROM staff
  S
    JOIN
    chef C ON C.sid = S.sid)t1
  GROUP BY t1.cooking_style)) t2
  ON chef.proficiency = t2.maxProficiency AND t2.cooking_style =
  chef.cooking_style)t3
  ON S.sid = t3.sid);
  ```

- Before modification

  ```
  SELECT * FROM staff;
  -- This is the table before raise the salary for best chef

   sid |       sname       | working_schdule | salary
  -----+-------------------+-----------------+--------
    18 | Melanie Hill      | evening         |     15 ->to modify
    17 | Luis Nash         | evening         |     15
    16 | Steve Marwan      | afternoon       |     15 ->to modify
  ```

```
  15 | Meadow Moran     | afternoon       |     15 ->to modify
  14 | Isla Greer       | morning         |     15 ->to modify
  13 | Lacie Barr       | morning         |     15 ->to modify
  12 | Ammie Summers    | evening         |     13
  11 | Lachlan Lawrence | evening         |     13
  10 | Cassie Noble     | afternoon       |     13
   9 | Jodie Stuart     | afternoon       |     13
   8 | Cara Allenr      | morning         |     13
   7 | Annie Carr       | evening         |     13
   6 | Hugo Turner      | evening         |     12
   5 | Anna Black       | evening         |     12
   4 | Jeffrey Frye     | afternoon       |     12
   3 | Macie Finely     | afternoon       |     12
   2 | Shala Tang       | morning         |     12
   1 | Samuel Randall   | morning         |     12
(18 rows)
```

- after modification

```
-- This is the table after raise the salary for best chef

 sid |      sname       | working_schdule | salary
-----+------------------+-----------------+--------
  18 | Melanie Hill     | evening         |     16 ->modified
  17 | Luis Nash        | evening         |     15
  16 | Steve Marwan     | afternoon       |     16 ->modified
  15 | Meadow Moran     | afternoon       |     16 ->modified
  14 | Isla Greer       | morning         |     16 ->modified
  13 | Lacie Barr       | morning         |     16 ->modified
  12 | Ammie Summers    | evening         |     13
  11 | Lachlan Lawrence | evening         |     13
  10 | Cassie Noble     | afternoon       |     13
   9 | Jodie Stuart     | afternoon       |     13
   8 | Cara Allenr      | evening         |     13
   7 | Annie Carr       | evening         |     13
   6 | Hugo Turner      | evening         |     12
   5 | Anna Black       | evening         |     12
   4 | Jeffrey Frye     | afternoon       |     12
   3 | Macie Finely     | afternoon       |     12
   2 | Shala Tang       | morning         |     12
   1 | Samuel Randall   | morning         |     12
(18 rows)
```

2. update the customer address

- Description

> When a customer want to update from old address to new address

- statement

```
UPDATE Customer
SET address='2020 Rue McGIll University'
WHERE canme='Armaan Gibbons';

SELECT address FROM Customer WHERE canme='Armaan Gibbons';
```

- before modification

```
SELECT canme, address FROM Customer WHERE canme='Armaan Gibbons';
-- This is the address before he/she change his delivery address

canme      |     address
----------------+------------------
 Armaan Gibbons | 3470 Rue Stanley
(1 row)
```

- after modification

```
-- This is the address after he/she change his delivery address

canme      |          address
----------------+----------------------------
 Armaan Gibbons | 2020 Rue McGIll University
(1 row)
```

3. A staff proposed to change his/her working schedule

- description

> A staff wants to change his/her working schedule. (eg. change from "morning" to "evening")

- statement

```
UPDATE staff
SET working_schdule = 'evening'
WHERE sid = 8;
```

- before modification

```
SELECT * FROM staff
ORDER BY sid;
-- This is the table before updating the working schedule
 sid |      sname       | working_schdule | salary
-----+------------------+-----------------+--------
   1 | Samuel Randall   | morning         |     12
   2 | Shala Tang       | morning         |     12
   3 | Macie Finely     | afternoon       |     12
   4 | Jeffrey Frye     | afternoon       |     12
   5 | Anna Black       | evening         |     12
   6 | Hugo Turner      | evening         |     12
   7 | Annie Carr       | evening         |     13
   8 | Cara Allenr      | morning         |     13 --> to modify
   9 | Jodie Stuart     | afternoon       |     13
  10 | Cassie Noble     | afternoon       |     13
  11 | Lachlan Lawrence | evening          |     13
  12 | Ammie Summers    | evening          |     13
  13 | Lacie Barr       | morning          |     16
  14 | Isla Greer       | morning          |     16
  15 | Meadow Moran     | afternoon        |     16
  16 | Steve Marwan     | afternoon        |     16
  17 | Luis Nash        | evening          |     15
  18 | Melanie Hill     | evening          |     16
(18 rows)
```

- after modification

```
-- This is the table after updating the working schedule
 sid |      sname       | working_schdule | salary
-----+------------------+-----------------+--------
   1 | Samuel Randall   | morning         |     12
   2 | Shala Tang       | morning         |     12
   3 | Macie Finely     | afternoon       |     12
   4 | Jeffrey Frye     | afternoon       |     12
   5 | Anna Black       | evening         |     12
   6 | Hugo Turner      | evening         |     12
   7 | Annie Carr       | evening         |     13
   8 | Cara Allenr      | evening         |     13 --> modified
   9 | Jodie Stuart     | afternoon       |     13
  10 | Cassie Noble     | afternoon       |     13
  11 | Lachlan Lawrence | evening         |     13
  12 | Ammie Summers    | evening         |     13
  13 | Lacie Barr       | morning         |     16
  14 | Isla Greer       | morning         |     16
  15 | Meadow Moran     | afternoon       |     16
  16 | Steve Marwan     | afternoon       |     16
  17 | Luis Nash        | evening         |     15
  18 | Melanie Hill     | evening         |     16
(18 rows)
```

4. raise the price of best seller dish

- description

> We select the best seller dish according to how many portion it was sold, and then we update the price

- statement

```
UPDATE dish
SET price = price * 1.1
WHERE dish_name IN
(SELECT dish_name FROM contain GROUP BY dish_name
HAVING COUNT(*) = (
SELECT MAX(y.dcount) FROM
(SELECT dish_name ,COUNT(*) AS dcount FROM contain GROUP BY dish_name) y));
```

- before modification

```
SELECT * FROM dish;

-- This is the table before we raise the price for bestseller dishes

              dish_name              | price
-------------------------------------+-------
 Sautéed Dark Beer Pork              |  28.9
 Simmered Peas & Mushroom Oysters    |  66.6
 Breaded Cucumber & Lime Pizza       |  19.9
 Roasted Almonds & Avocado Bread     |  28.9
 Rum and Praline Delight             |  17.9
 Chestnut and Nutmeg Gingerbread     |  25.9
 Ginger Candy                        |  5.99
 Cranberry Genoise                   |    27
 Fire-Roasted Basil & Mint Yak       |  32.3
 Simmered Mountain Rabbit            |  22.5
 Pressure-Fried Vegetables & Frog    |  31.5
 Sautéed Orange & Mustard Vegetables |  26.9
 Barbecued Mustard & Garlic Calzone  |    25
 boiled spicy fish                   | 30.99
 Guoyou pork                         | 17.99
 Sweet and sour pork ribs            |  14.5
 beef pho                            |  10.5
 General Tsos Chicken                |  15.5
 Tenderized Truffles & Yak           | 50.99 -> to modify
 Tenderized Parmesan Lobster         |  99.9 -> to modify
(20 rows)
```

- after modification

```
SELECT * FROM dish;
-- This is the table after we raise the price for bestseller dishes

              dish_name              | price
-------------------------------------+--------
 Sautéed Dark Beer Pork              |   28.9
 Simmered Peas & Mushroom Oysters    |   66.6
 Breaded Cucumber & Lime Pizza       |   19.9
 Roasted Almonds & Avocado Bread     |   28.9
 Rum and Praline Delight             |   17.9
 Chestnut and Nutmeg Gingerbread     |   25.9
 Ginger Candy                        |   5.99
 Cranberry Genoise                   |     27
 Fire-Roasted Basil & Mint Yak       |   32.3
 Simmered Mountain Rabbit            |   22.5
 Pressure-Fried Vegetables & Frog    |   31.5
 Sautéed Orange & Mustard Vegetables |   26.9
 Barbecued Mustard & Garlic Calzone  |     25
 boiled spicy fish                   |  30.99
 Guoyou pork                         |  17.99
 Sweet and sour pork ribs            |   14.5
 beef pho                            |   10.5
 General Tsos Chicken                |   15.5
 Tenderized Truffles & Yak           | 56.089 -> modified
 Tenderized Parmesan Lobster         | 109.89 -> modified
(20 rows)
```

Q7. create views

Best_chef

- Create view

```
CREATE VIEW bestChef (sname, cooking_style)
AS
SELECT S.sname, t3.cooking_style FROM staff S JOIN
(SELECT chef.sid, chef.proficiency,chef.cooking_style FROM chef
INNER JOIN
((SELECT t1.cooking_style, MAX(t1.proficiency) AS maxProficiency FROM
  (SELECT DISTINCT S.sname, S.sid, C.proficiency, C.cooking_style FROM staff
S
  JOIN
  chef C ON C.sid = S.sid)t1
GROUP BY t1.cooking_style)) t2
ON chef.proficiency = t2.maxProficiency AND t2.cooking_style =
chef.cooking_style)t3
ON S.sid = t3.sid;
```

- result

```
    sname     | sid | proficiency | cooking_style
--------------+-----+-------------+---------------
 Lacie Barr   |  13 |           2 | Cutting
 Isla Greer   |  14 |           3 | Steaming
 Meadow Moran |  15 |           4 | Braising
 Steve Marwan |  16 |           4 | Braising
 Melanie Hill |  18 |           5 | Stir-frying
(5 rows)
```

- description

```
We create view to select the best chef on each of the cooking styles. we can
post this information of our best chef to customers using this view.
```

- Query

```
UPDATE best_chef
SET proficiency = 5
WHERE sid = 5;
```

- error

```
ERROR:  cannot update view "best_chef"
DETAIL:  Views that do not select from a single table or view are not
automatically updatable.
HINT:  To enable updating the view, provide an INSTEAD OF UPDATE trigger or
an unconditional ON UPDATE DO INSTEAD rule.
SQL state: 55000
```

- explanation: Views are not selected from a single table so it cannot be updated

Delivery_order_info

- create view

```
CREATE VIEW delivery_order_info(order_number, delivery_guy_name,
delivery_guy_sid,
  delivery_guy_phone, cus_name, cus_phone, cus_address)
AS
SELECT t2.order_number, s.sname AS delivery_guy_name, t2.sid AS
delivery_guy_sid,
  t2.delivery_guy_phone, t2.cus_name, t2.cus_phone, t2.cus_address
FROM staff s
INNER JOIN
  (
    SELECT t1.order_number, t1.cus_name, t1.cus_phone, t1.cus_address,
      guy.phone_number AS delivery_guy_phone, t1.sid AS sid
    FROM delivery_guy guy
    INNER JOIN
    (
      SELECT cus.phone_number AS cus_phone, cus.address AS cus_address,
        cus.canme AS cus_name, sid, order_number
      FROM customer cus INNER JOIN delivery_orders d
      ON cus.phone_number = d.phone_number
    ) t1
    ON t1.sid = guy.sid
  ) t2
ON s.sid = t2.sid;
```

- result

```
order_number | delivery_guy_name | delivery_guy_sid | delivery_guy_phone |
cus_name    |  cus_phone   |       cus_address
--------------+-------------------+------------------+--------------------+-
------------+--------------+--------------------------
 006          | Annie Carr        |                7 | 514-999-9999       |
Ava Pollard | 514-202-1544 | 1120  th Avenue
 007          | Jodie Stuart      |                9 | 514-777-7777       |
Teegan Liu  | 514-303-2334 | 3417  Brew Creek Rd
 008          | Cassie Noble      |               10 | 514-666-6666       |
Jing Guo    | 514-404-2154 | 283  Jasper Avenue
 009          | Jodie Stuart      |                9 | 514-777-7777       |
Rong Huang  | 514-273-3594 | 3242  Boulevard Cremazie
 014          | Cassie Noble      |               10 | 514-666-6666       |
Rong Huang  | 514-273-3594 | 3242  Boulevard Cremazie
 015          | Cara Allenr       |                8 | 514-888-8888       |
Teegan Liu  | 514-303-2334 | 3417  Brew Creek Rd
(6 rows)
```

- description

```
We create view to related info between customers and delivery orders. We can
post this information of the delivery to customer.
```

- query

```
UPDATE delivery_order_info
SET cus_address = '1430 rue citycouncillors'
WHERE order_number = '006';
```

- error

```
ERROR:  cannot update view "delivery_order_info"
DETAIL:  Views that do not select from a single table or view are not
automatically updatable.
HINT:  To enable updating the view, provide an INSTEAD OF UPDATE trigger or
an unconditional ON UPDATE DO INSTEAD rule.
SQL state: 55000
```

- explanation: Views are not selected from a single table so it cannot be updated

---

Q8. `CHECK` Constraints

Constraint 1

Description:

> The `proficiency` of the `chef` should between 0 and 5.

Add Constraint (SQL statement):

```
ALTER TABLE chef ADD CONSTRAINT proficiency CHECK (proficiency >= 0 AND
proficiency <= 5);
```

Response:

```
ALTER TABLE
```

Revised Schema:

```
Table "cs421g62.chef"
    Column     |         Type          | Modifiers
---------------+-----------------------+-----------
 proficiency   | integer               |
 cooking_style | character varying(12) |
 sid           | integer               | not null
Indexes:
    "chef_pkey" PRIMARY KEY, btree (sid)
Check constraints:
    "proficiency" CHECK (proficiency >= 0 AND proficiency <= 5)
Foreign-key constraints:
    "chef_sid_fkey" FOREIGN KEY (sid) REFERENCES staff(sid)
Referenced by:
    TABLE "cooked_by" CONSTRAINT "cooked_by_sid_fkey" FOREIGN KEY (sid)
REFERENCES chef(sid)
```

Try to violate the constraint:

```
cs421=> UPDATE chef SET proficiency = 6 WHERE sid = 18;
```

Response:

```
ERROR:  new row for relation "chef" violates check constraint "proficiency"
DETAIL:  Failing row contains (6, Stir-frying, 18).
```

Constraint 2

Description:

> The `working schedule` of the `staff` should be in morning, afternoon or evening.

Add Constraint (SQL statement):

```
ALTER TABLE staff ADD CONSTRAINT working_schdule CHECK (working_schdule IN ('mor
ning', 'afternoon', 'evening'));
```

Response:

```
ALTER TABLE
```

Revised Schema:

```
cs421=> \d staff
               Table "cs421g62.staff"
```

```
     Column      |         Type          | Modifiers
-----------------+-----------------------+-----------
 sid             | integer               | not null
 sname           | character varying(40) |
 working_schdule | character varying(12) |
 salary          | double precision      |
Indexes:
    "staff_pkey" PRIMARY KEY, btree (sid)
Check constraints:
    "working_schdule" CHECK (working_schdule::text = ANY
(ARRAY['morning'::character varying, 'afternoon'::character varying,
'evening'::character varying]::text[]))
Referenced by:
    TABLE "chef" CONSTRAINT "chef_sid_fkey" FOREIGN KEY (sid) REFERENCES
staff(sid)
    TABLE "delivery_guy" CONSTRAINT "delivery_guy_sid_fkey" FOREIGN KEY (sid)
REFERENCES staff(sid)
    TABLE "delivery_orders" CONSTRAINT "delivery_orders_sid_fkey" FOREIGN KEY
(sid) REFERENCES staff(sid)
    TABLE "served_by" CONSTRAINT "served_by_sid_fkey" FOREIGN KEY (sid)
REFERENCES staff(sid)
    TABLE "waiter" CONSTRAINT "waiter_sid_fkey" FOREIGN KEY (sid) REFERENCES
staff(sid)
```

Try to violate the constraint:

```
cs421=> UPDATE staff  SET working_schdule  = 'mid_night'  WHERE sid = 18;
```

Response:

```
ERROR:  new row for relation "staff" violates check constraint "working_schdule"
DETAIL:  Failing row contains (18, Melanie Hill, mid_night, 16).
```

Q9. Creativity

Real Data Sets

We used https://mockaroo.com/ to generate data which are more real world like. We generated 400 real world like customers. (The data generating SQL file is called `customer.sql`)

A few example of auto-generated statements:

```
insert into customer (phone_number, canme, address) values ('237-431-2785',
'Maurie Strutz', '85 Di Loreto Circle');
insert into customer (phone_number, canme, address) values ('179-115-9035',
'Kore Peerman', '865 Spaight Place');
insert into customer (phone_number, canme, address) values ('231-819-7747',
'Marilyn Godard', '241 Crescent Oaks Center');
insert into customer (phone_number, canme, address) values ('598-838-8001',
'Webster Betteney', '95 Harper Road');
insert into customer (phone_number, canme, address) values ('642-540-0806',
'Petunia Longden', '68657 Miller Center');
insert into customer (phone_number, canme, address) values ('340-569-5162',
'Yank Gurdon', '7554 Holy Cross Alley');
............
```

Result:

```
cs421=> SELECT COUNT(*) AS number_records From customer;
 number_records
----------------
            409
(1 row)
```

Complex Analytical Queries

Description:

> Select best `chef` for each of the `cooking style`

SQL query:

```
SELECT S.sname, t3.proficiency, t3.cooking_style,S.sid FROM staff S JOIN
(SELECT chef.sid, chef.proficiency,chef.cooking_style FROM chef
INNER JOIN
(
    (
        SELECT t1.cooking_style, MAX(t1.proficiency) AS maxProficiency
        FROM (SELECT DISTINCT S.sname, S.sid, C.proficiency, C.cooking_style
FROM staff S
        JOIN chef C ON C.sid = S.sid) t1
        GROUP BY t1.cooking_style
    )
) t2
ON chef.proficiency = t2.maxProficiency AND t2.cooking_style =
chef.cooking_style)t3
ON S.sid = t3.sid;
```

Result:

```
    sname     | proficiency | cooking_style
--------------+-------------+---------------
 Lacie Barr   |           2 | Cutting
 Isla Greer   |           3 | Steaming
 Meadow Moran |           4 | Braising
 Steve Marwan |           4 | Braising
 Melanie Hill |           5 | Stir-frying
(5 rows)
```

Complex Business Requirement:

- This could select all the best chef for each of the cooking style
- We could raise the salary for those best-performing chefs to encourage all the chefs work harder
- Increase the salary of chef with the highest rating

SQL for increasing the salary of best-performing chef:

```
UPDATE staff
SET salary = salary+1
WHERE sid IN
(SELECT S.sid FROM staff S JOIN
(SELECT chef.sid, chef.proficiency,chef.cooking_style FROM chef
INNER JOIN
((SELECT t1.cooking_style, MAX(t1.proficiency) AS maxProficiency FROM
  (SELECT DISTINCT S.sname, S.sid, C.proficiency, C.cooking_style FROM staff S
  JOIN
  chef C ON C.sid = S.sid)t1
GROUP BY t1.cooking_style)) t2
ON chef.proficiency = t2.maxProficiency AND t2.cooking_style =
chef.cooking_style)t3
ON S.sid = t3.sid);
```

- Before modification

```
 sid |       sname       | working_schdule | salary
-----+-------------------+-----------------+--------
  18 | Melanie Hill      | evening         |     15 ->to modify
  17 | Luis Nash         | evening         |     15
  16 | Steve Marwan      | afternoon       |     15 ->to modify
  15 | Meadow Moran      | afternoon       |     15 ->to modify
  14 | Isla Greer        | morning         |     15 ->to modify
  13 | Lacie Barr        | morning         |     15 ->to modify
```

- after modification

```
 sid |       sname       | working_schdule | salary
-----+-------------------+-----------------+--------
  18 | Melanie Hill      | evening         |     16 ->modified
  17 | Luis Nash         | evening         |     15
  16 | Steve Marwan      | afternoon       |     16 ->modified
  15 | Meadow Moran      | afternoon       |     16 ->modified
  14 | Isla Greer        | morning         |     16 ->modified
  13 | Lacie Barr        | morning         |     16 ->modified
```