Database Principle and Design
CSC-420-01-242

2/16/2025
Project

Clay Jones
Isaiah Boyd
Wendon Doswell

Part 1 Proposal

Database

a. Description:

Our database will be based on an HBCU cafeteria repository. It will include multiple HBCU dining areas and store information about their menus. Categories will include the name of the building, serving amount, day of week, time of day, main course, first side, second side, dessert, and the special drink. This database will store all the planned meals for the week at any given time. This database will contain multiple different relationships however we will not know the exact relationships until we normalize the data. Through normalization, we will ensure that the data is structured efficiently to minimize redundancy and maintain data integrity.

b. Inspiration:

Our inspiration from this project was seeing how Norfolk State University has a dining services website that shows the menu for each week. Seeing how the cafe has an effective way to track the menu, we thought it would be a good idea. In order to do this, they store the dinner items in a database and it updates weekly. We wanted to do this because it can be implemented at different institutions.

c. Questions:

How much staff is available on Thursdays?
What is the cafeteria serving on Wednesday?
What cafeteria is serving chicken tenders?
What does the cafeteria serve on Monday mornings?
What drink is at the cafeteria today?

d. Reports List

1. Allergen/Ingredients - A report which list which specific foods contain certain ingredients such peppers, salt, cheese, nuts

2. Options -A report that shows which foods are a possible vegetarian or vegan option rather than the general option

3. Meal popularity - A report which tracks which dishes are most frequently served and preferred by students.

e. Business Rules

1. **Each meal must have a list of sides associated with it.**

a. A meal cannot be stored in the database without its side.
b. Each side must be linked to at least one meal to ensure accurate inventory tracking.

2. **A dish may be marked as vegetarian, vegan, or gluten-free, but must meet the dietary requirements to be classified as such.**

    a. A dish labeled vegetarian cannot contain meat, while a vegan dish must exclude all animal products, including dairy and eggs.
    b. Gluten-free dishes must not contain wheat, barley, rye, or any gluten-containing additives.

3. **Meals must be planned and stored in the database at least one week in advance so the menu can be created.**

    a. The database should prevent the creation of a menu for a given week if the meals have not been planned in advance.
    b. Any changes to a planned meal must be made at least 24 hours before the scheduled serving time to ensure proper preparation.

4. All dining areas must be assigned a unique identifier and store relevant information.

    a. A dining area must have a unique ID, name, and location in the database.
    b. A record of the capacity and staffing details of each dining area must be maintained to optimize operations.

5. All students and faculty must have access to view weekly menus through a user-friendly interface.

    a. A digital menu must be available for users to check meal options for the week.
    b. A filtering system should allow users to search for meals based on dietary preferences (e.g., vegetarian, vegan, gluten-free).

f. Possible Pitfall

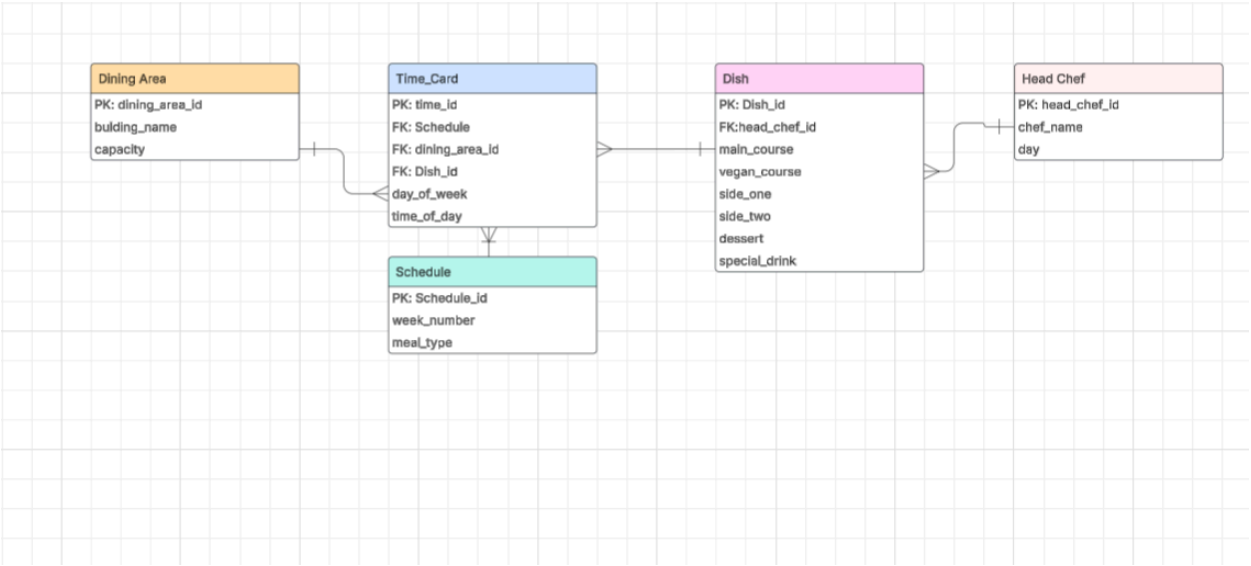
The amount of ingredients in a dish may be too large to include for an entire meal so only the main ones might have to be displayed.

Database Principle and Design
CSC-420-01-242

4/10/2025
Project

Clay Jones
Isaiah Boyd
Wendon Doswell

Part 2 Model

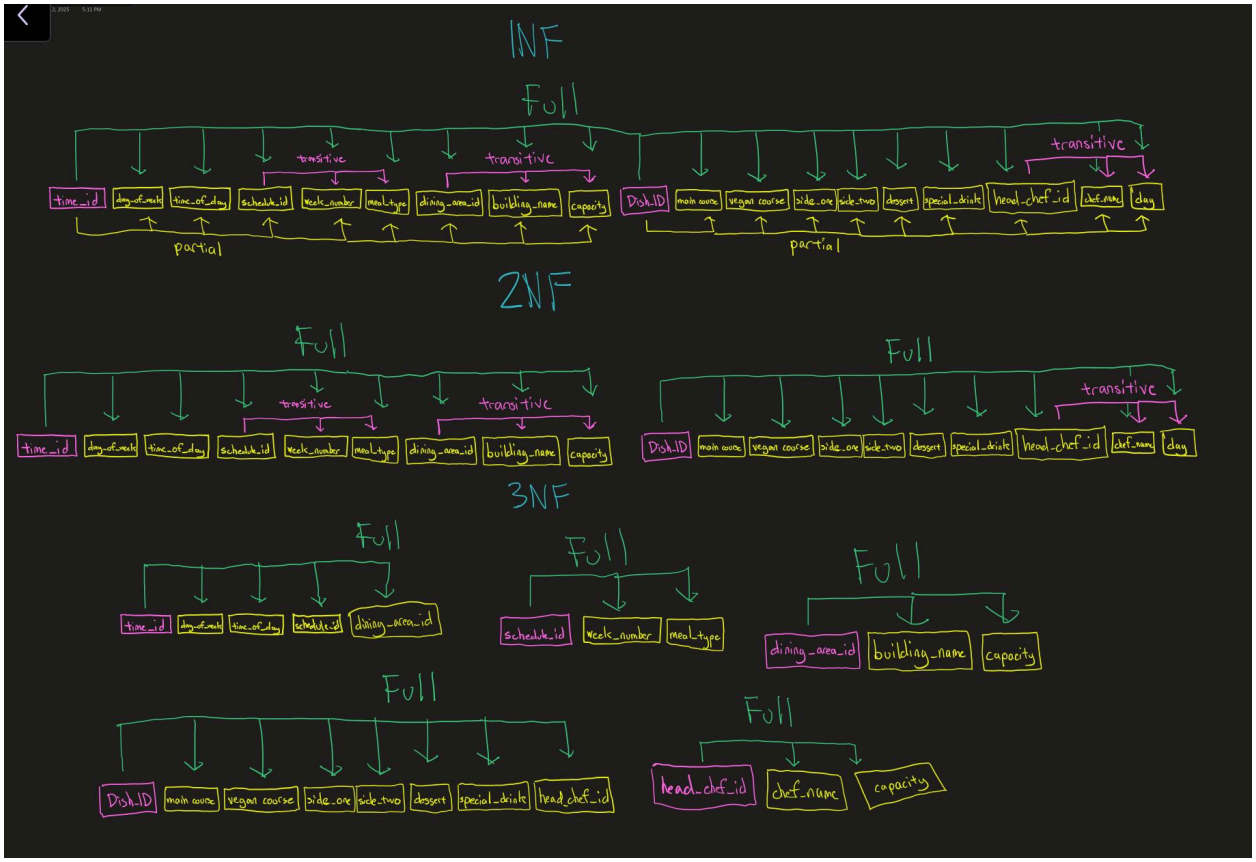

Dining Area
PK: dining_area_id
bulding_name
capacity
Time_Card
PK: time_id
FK: Schedule
FK: dining_area_id
FK: Dish_id
day_of_week
time_of_day
Schedule
PK: Schedule_id
week_number
meal_type
Dish
PK: Dish_id
FK:head_chef_id
main_course
vegan_course
side_one
side_two
dessert
special_drink
Head Chef
PK: head_chef_id
chef_name
day

Database Principle and Design
CSC-420-01-242

4/10/2025
Project

Clay Jones
Isaiah Boyd
Wendon Doswell

Part 3 Normalization


1NF
Full
transitive
transitive
transitive
time_id
day_of_week
time_of_day
schedule_id
week_number
meal_type
dining_area_id
building_name
capacity
Dish_ID
main course
vegan course
side_one
side_two
dessert
special_drink
head_chef_id
chef_name
day
partial
partial
2NF
Full
transitive
transitive
Full
transitive
3NF
Full
Full
Full
Full
Full

Database Principle and Design
CSC-420-01-242

4/10/2025
Project

Clay Jones
Isaiah Boyd
Wendon Doswell

Data Dictionary

Data Dictionary

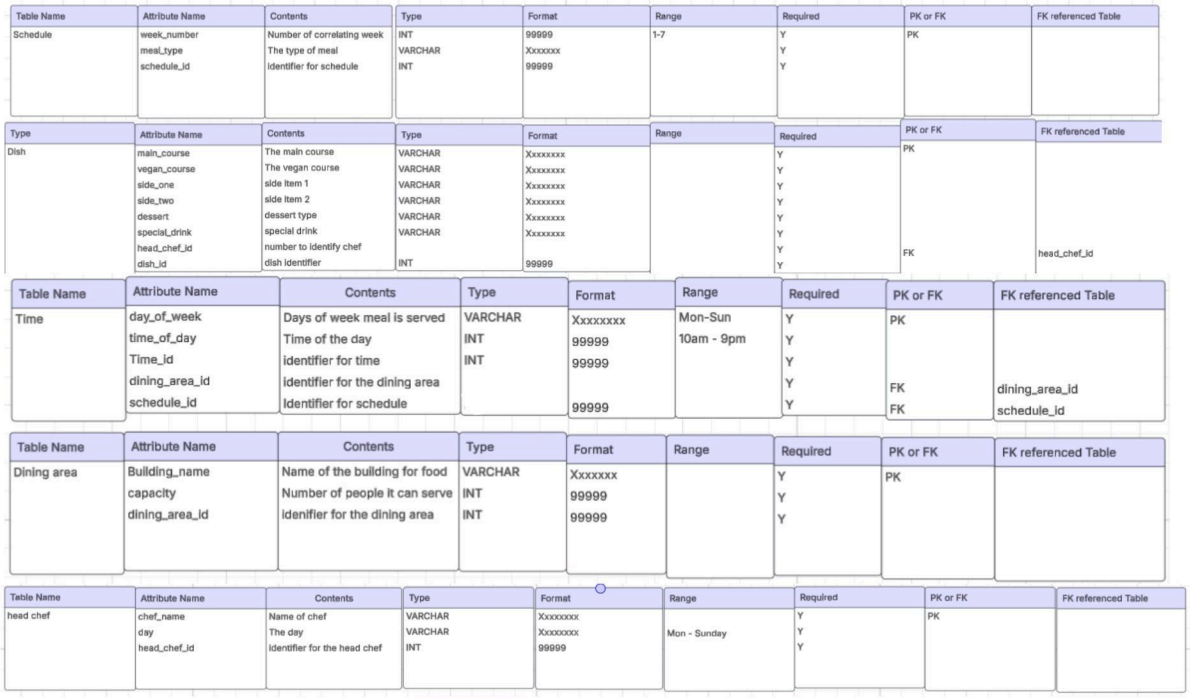


Table Name	Attribute Name	Contents	Type	Format	Range	Required	PK or FK	FK referenced Table
Schedule	week_number	Number of correlating week	INT	99999	1-7	Y	PK	
	meal_type	The type of meal	VARCHAR	Xxxxxx		Y		
	schedule_id	identifier for schedule	INT	99999		Y		

Type	Attribute Name	Contents	Type	Format	Range	Required	PK or FK	FK referenced Table
Dish	main_course	The main course	VARCHAR	Xxxxxxx		Y	PK	
	vegan_course	The vegan course	VARCHAR	Xxxxxxx		Y		
	side_one	side item 1	VARCHAR	Xxxxxxx		Y		
	side_two	side item 2	VARCHAR	Xxxxxxx		Y		
	dessert	dessert type	VARCHAR	Xxxxxxx		Y		
	special_drink	special drink	VARCHAR	Xxxxxxx		Y		
	head_chef_id	number to identify chef				Y	FK	head_chef_id
	dish_id	dish identifier	INT	99999		Y		

Table Name	Attribute Name	Contents	Type	Format	Range	Required	PK or FK	FK referenced Table
Time	day_of_week	Days of week meal is served	VARCHAR	Xxxxxxxx	Mon-Sun	Y	PK	
	time_of_day	Time of the day	INT	99999	10am - 9pm	Y		
	Time_id	identifier for time	INT	99999		Y		
	dining_area_id	identifier for the dining area				Y	FK	dining_area_id
	schedule_id	identifier for schedule		99999		Y	FK	schedule_id

Table Name	Attribute Name	Contents	Type	Format	Range	Required	PK or FK	FK referenced Table
Dining area	Building_name	Name of the building for food	VARCHAR	Xxxxxxx		Y	PK	
	capacity	Number of people it can serve	INT	99999		Y		
	dining_area_id	idenifier for the dining area	INT	99999		Y		

Table Name	Attribute Name	Contents	Type	Format	Range	Required	PK or FK	FK referenced Table
head chef	chef_name	Name of chef	VARCHAR	Xxxxxxx		Y	PK	
	day	The day	VARCHAR	Xxxxxxx	Mon - Sunday	Y		
	head_chef_id	identifier for the head chef	INT	99999		Y		

Database Principle and Design
CSC-420-01-242

4/28/2025
Project

Clay Jones
Isaiah Boyd
Wendon Doswell

Part 4 Create DB

Our sql database code along with data population:

```
-- Start logging output
SPOOL output.txt

-- Drop existing tables
DROP TABLE time CASCADE CONSTRAINTS;
DROP TABLE dish CASCADE CONSTRAINTS;
DROP TABLE schedule CASCADE CONSTRAINTS;
DROP TABLE dining_area CASCADE CONSTRAINTS;
DROP TABLE head_chef CASCADE CONSTRAINTS;

-- ======================================
-- TABLE: Head_Chef
-- ======================================
CREATE TABLE head_chef (
    head_chef_id INT PRIMARY KEY,
    chef_name VARCHAR2(50) NOT NULL
);

-- ======================================
-- TABLE: Dish (References: Head_Chef)
-- ======================================
CREATE TABLE dish (
    dish_id INT PRIMARY KEY,
    main_course VARCHAR2(50) NOT NULL,
    vegan_course VARCHAR2(50) NOT NULL,
    side_one VARCHAR2(50) NOT NULL,
    side_two VARCHAR2(50) NOT NULL,
    dessert VARCHAR2(50) NOT NULL,
    special_drink VARCHAR2(50) NOT NULL,
    head_chef_id INT,
    CONSTRAINT fk_dish_headchef FOREIGN KEY (head_chef_id) REFERENCES
head_chef(head_chef_id)
);

-- ======================================
-- TABLE: Schedule
-- ======================================
CREATE TABLE schedule (
    schedule_id INT PRIMARY KEY,
    meal_type VARCHAR2(20) NOT NULL CHECK (meal_type IN ('Breakfast', 'Lunch', 'Dinner')),
    week_number INT CHECK (week_number BETWEEN 1 AND 7)
);
```

```
-- ======================================
-- TABLE: Dining_Area
-- ======================================
CREATE TABLE dining_area (
    dining_area_id INT PRIMARY KEY,
    building_name VARCHAR2(50) NOT NULL,
    capacity INT NOT NULL
);

-- ======================================
-- TABLE: Time (References: Dining_Area, Schedule)
-- ======================================
CREATE TABLE time (
    time_id INT PRIMARY KEY,
    day_of_week VARCHAR2(10) NOT NULL CHECK (day_of_week IN ('Mon', 'Tue', 'Wed',
'Thu', 'Fri', 'Sat', 'Sun')),
    time_category VARCHAR2(15) NOT NULL CHECK (time_category IN ('Morning', 'Afternoon',
'Evening')),
    dining_area_id INT NOT NULL,
    schedule_id INT NOT NULL,
    CONSTRAINT fk_time_dining FOREIGN KEY (dining_area_id) REFERENCES
dining_area(dining_area_id),
    CONSTRAINT fk_time_schedule FOREIGN KEY (schedule_id) REFERENCES
schedule(schedule_id)
);

-- ======================================
-- Insert Records
-- ======================================

-- Insert into Head_Chef
INSERT INTO head_chef VALUES (1, 'Alice Johnson');
INSERT INTO head_chef VALUES (2, 'Brandon Lee');
INSERT INTO head_chef VALUES (3, 'Isaiah Boyd');
INSERT INTO head_chef VALUES (4, 'Wendon Doswell');
INSERT INTO head_chef VALUES (5, 'Clay Jones');

-- Insert into Dish
INSERT INTO dish VALUES (1, 'Grilled Chicken', 'Tofu Stir Fry', 'Mashed Potatoes', 'Steamed
Broccoli', 'Chocolate Cake', 'Berry Smoothie', 1);
INSERT INTO dish VALUES (2, 'Beef Lasagna', 'Vegan Pasta', 'Garlic Bread', 'Garden Salad',
'Cheesecake', 'Lemonade', 2);
```

```
INSERT INTO dish VALUES (3, 'Fish Tacos', 'Black Bean Tacos', 'Rice Pilaf', 'Corn on the Cob',
'Apple Pie', 'Iced Tea', 3);
INSERT INTO dish VALUES (4, 'Turkey Sandwich', 'Veggie Wrap', 'Potato Chips', 'Coleslaw',
'Brownies', 'Sweet Tea', 4);
INSERT INTO dish VALUES (5, 'Steak', 'Grilled Veggies', 'Baked Potato', 'Green Beans',
'Banana Pudding', 'Orange Juice', 5);

-- Insert into Schedule
INSERT INTO schedule VALUES (1, 'Breakfast', 1);
INSERT INTO schedule VALUES (2, 'Lunch', 2);
INSERT INTO schedule VALUES (3, 'Dinner', 3);
INSERT INTO schedule VALUES (4, 'Lunch', 4);
INSERT INTO schedule VALUES (5, 'Dinner', 5);

-- Insert into Dining_Area
INSERT INTO dining_area VALUES (1, 'Oak Hall', 150);
INSERT INTO dining_area VALUES (2, 'Pine Commons', 200);
INSERT INTO dining_area VALUES (3, 'Maple Dining', 120);
INSERT INTO dining_area VALUES (4, 'Elm Bistro', 80);
INSERT INTO dining_area VALUES (5, 'Cedar Cafe', 100);

-- Insert into Time
INSERT INTO time VALUES (1, 'Mon', 'Morning', 1, 1);
INSERT INTO time VALUES (2, 'Tue', 'Afternoon', 2, 2);
INSERT INTO time VALUES (3, 'Wed', 'Evening', 3, 3);
INSERT INTO time VALUES (4, 'Thu', 'Afternoon', 4, 4);
INSERT INTO time VALUES (5, 'Fri', 'Evening', 5, 5);

-- Stop logging output
SPOOL OFF;
```

Query Outputs

1. Write a Select all query for all tables in your database

```
SQL> SELECT * FROM head_chef;

  HEAD_CHEF_ID        CHEF_NAME
______________ ____________________
             1 Alice Johnson
             2 Brandon Lee
             3 Isaiah Boyd
             4 Wendon Doswell
             5 Clay Jones

SQL> SELECT * FROM dish;

  DISH_ID       MAIN_COURSE       VEGAN_COURSE          SIDE_ONE
SIDE_TWO          DESSERT     SPECIAL_DRINK    HEAD_CHEF_ID
__________ ____________________ ____________________ ____________________
__________                                                          ________
____________________ ____________________ ____________________
         1 Grilled Chicken    Tofu Stir Fry       Mashed Potatoes    Steamed Bro
ccoli    Chocolate Cake    Berry Smoothie                 1
         2 Beef Lasagna       Vegan Pasta         Garlic Bread       Garden Sala
d        Cheesecake        Lemonade                       2
         3 Fish Tacos         Black Bean Tacos    Rice Pilaf         Corn on the
Cob     Apple Pie         Iced Tea                       3
         4 Turkey Sandwich    Veggie Wrap         Potato Chips       Coleslaw
Brownies          Sweet Tea                      4
         5 Steak              Grilled Veggies     Baked Potato       Green Beans
Banana Pudding    Orange Juice                   5

SQL> SELECT * FROM schedule;

  SCHEDULE_ID    MEAL_TYPE    WEEK_NUMBER
______________ _____________ _____________
             1 Breakfast                1
             2 Lunch                    2
             3 Dinner                   3
             4 Lunch                    4
             5 Dinner                   5

SQL> SELECT * FROM dining_area;
```

DINING_AREA_ID	BUILDING_NAME	CAPACITY
1	Oak Hall	150
2	Pine Commons	200
3	Maple Dining	120
4	Elm Bistro	80
5	Cedar Cafe	100

SQL> SELECT * FROM time;

TIME_ID	DAY_OF_WEEK	TIME_CATEGORY	DINING_AREA_ID	SCHEDULE_ID
1	Mon	Morning	1	1
2	Tue	Afternoon	2	2
3	Wed	Evening	3	3
4	Thu	Afternoon	4	4
5	Fri	Evening	5	5

```
SQL> SELECT * FROM head_chef;

   HEAD_CHEF_ID         CHEF_NAME
_______________ _________________
              1 Alice Johnson
              2 Brandon Lee
              3 Isaiah Boyd
              4 Wendon Doswell
              5 Clay Jones

SQL> SELECT * FROM dish;

   DISH_ID        MAIN_COURSE        VEGAN_COURSE           SIDE_ONE
SIDE_TWO           DESSERT     SPECIAL_DRINK    HEAD_CHEF_ID
_________ _________________ _________________ ________________ __________
________ _________________ _________________ _______________
         1 Grilled Chicken    Tofu Stir Fry      Mashed Potatoes    Steamed Bro
ccoli    Chocolate Cake    Berry Smoothie                  1
         2 Beef Lasagna       Vegan Pasta        Garlic Bread       Garden Sala
d        Cheesecake        Lemonade                        2
         3 Fish Tacos         Black Bean Tacos   Rice Pilaf         Corn on the
 Cob     Apple Pie         Iced Tea                        3
         4 Turkey Sandwich    Veggie Wrap        Potato Chips       Coleslaw
         Brownies          Sweet Tea                       4
         5 Steak              Grilled Veggies    Baked Potato       Green Beans
         Banana Pudding    Orange Juice                    5

SQL> SELECT * FROM schedule;

   SCHEDULE_ID    MEAL_TYPE    WEEK_NUMBER
______________ ____________ _____________
             1 Breakfast                  1
             2 Lunch                      2
             3 Dinner                     3
             4 Lunch                      4
             5 Dinner                     5
```

```
SQL> SELECT * FROM dining_area;

   DINING_AREA_ID    BUILDING_NAME    CAPACITY
_________________ _________________ __________
                1 Oak Hall                 150
                2 Pine Commons             200
                3 Maple Dining             120
                4 Elm Bistro                80
                5 Cedar Cafe               100

SQL> SELECT * FROM time;

   TIME_ID    DAY_OF_WEEK    TIME_CATEGORY    DINING_AREA_ID    SCHEDULE_ID
_________ _____________ _____________ _________________ _____________
        1 Mon           Morning                       1               1
        2 Tue           Afternoon                     2               2
        3 Wed           Evening                       3               3
        4 Thu           Afternoon                     4               4
        5 Fri           Evening                       5               5
```

2. Write a query which uses the count function (you select the table)

```
 TOTAL_DISHES
_____________
           5
```

```
SQL> SELECT COUNT(*) AS total_dishes FROM dish;

   TOTAL_DISHES
_______________
              5
```

3. Write a query which uses the Where clause

DISH_ID	MAIN_COURSE	VEGAN_COURSE	SIDE_ONE	SIDE_TWO	DESSERT	SPECIAL_DRINK	HEAD_CHEF_ID
2	Beef Lasagna	Vegan Pasta	Garlic Bread	Garden Salad	Cheesecake	Lemonade	2

```
SQL> SELECT * FROM dish
  2  WHERE head_chef_id = 2;

   DISH_ID     MAIN_COURSE    VEGAN_COURSE       SIDE_ONE       SIDE_TWO       DESSERT    SPECIAL_DRINK    HEAD_CHEF_ID
__________ _______________ _______________ ______________ ______________ _____________ ________________ _____________
         2 Beef Lasagna    Vegan Pasta     Garlic Bread   Garden Salad   Cheesecake    Lemonade                    2
```

4. Write a query which uses the Group By Clause

MEAL_TYPE	TOTAL_MEALS
Lunch	2
Dinner	2
Breakfast	1

```
SQL> SELECT meal_type, COUNT(*) AS total_meals
  2  FROM schedule
  3  GROUP BY meal_type;

   MEAL_TYPE    TOTAL_MEALS
_____________ _______________
Lunch                     2
Dinner                    2
Breakfast                 1
```

5. Write a query which uses the Having Clause

MEAL_TYPE	TOTAL_MEALS
Lunch	2
Dinner	2

```
SQL> SELECT meal_type, COUNT(*) AS total_meals
  2  FROM schedule
  3  GROUP BY meal_type
  4  HAVING COUNT(*) > 1;

   MEAL_TYPE    TOTAL_MEALS
____________ _____________
Lunch                    2
Dinner                   2
```

6. Write a query which uses a Natural Join

SCHEDULE_ID	TIME_ID	DAY_OF_WEEK	TIME_CATEGORY	DINING_AREA_ID	MEAL_TYPE	WEEK_NUMBER
1	1	Mon	Morning	1	Breakfast	1
2	2	Tue	Afternoon	2	Lunch	2
3	3	Wed	Evening	3	Dinner	3
4	4	Thu	Afternoon	4	Lunch	4
5	5	Fri	Evening	5	Dinner	5

```
SQL> SELECT *
  2  FROM time
  3  NATURAL JOIN schedule;

   SCHEDULE_ID    TIME_ID    DAY_OF_WEEK    TIME_CATEGORY    DINING_AREA_ID    MEAL_TYPE    WEEK_NUMBER
______________ __________ ______________ ________________ _________________ ____________ _____________
           1          1 Mon            Morning                          1 Breakfast                1
           2          2 Tue            Afternoon                        2 Lunch                    2
           3          3 Wed            Evening                          3 Dinner                   3
           4          4 Thu            Afternoon                        4 Lunch                    4
           5          5 Fri            Evening                          5 Dinner                   5
```

7. Write a sub query

CHEF_NAME
Alice Johnson

```
SQL> SELECT chef_name
  2  FROM head_chef
  3  WHERE head_chef_id IN (
  4      SELECT head_chef_id
  5      FROM dish
  6      WHERE dessert LIKE '%Cake%'
  7  );

      CHEF_NAME
________________
Alice Johnson
```

Division of Work:
Clay - creating code for the database
Isaiah - populating the database and adding describe statements
Wendon - creating and inserting queries