Flavortown

BMGT402 Database Design

Gemma Nguyen, Arielle Bernstein, Jack McDonough,
Hibah Khan, Zachary Wolf

Mall Food Court Database Analysis

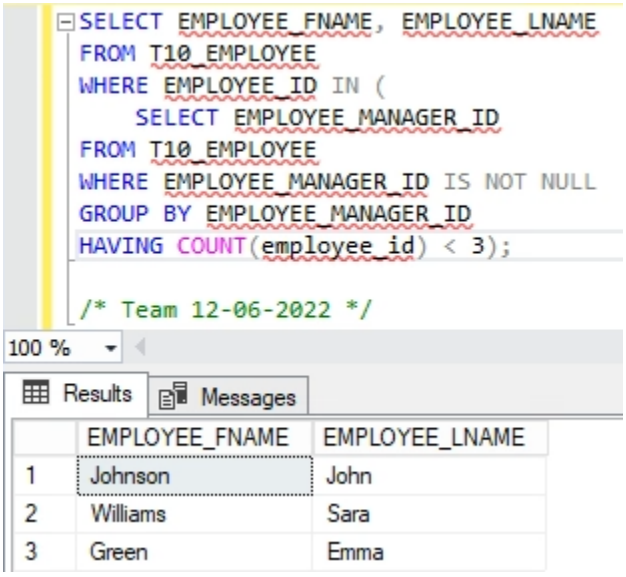
Mission Statement:
Our goal is to develop a database that can help us track customers and orders within the entire food court. This will allow the restaurants to staff employees around peak days and times as well as allow the mall to produce targeted marketing material for their best performing menu items.

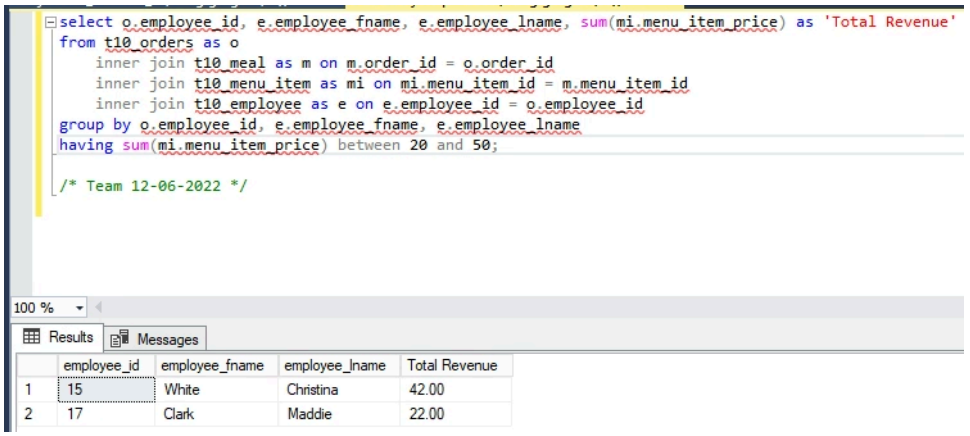
Mission Objectives:

1. Determine which restaurants are under staffed
2. Determine which employees are bringing in the most revenue
3. Determine which menu items are considered to be premium based on its pricing
4. Identify which menu items are considered cheapest based on pricing
5. Identify which menu options are popular
6. Identify which restaurants are the most popular
7. Determine the number of orders placed within a given season of the year
8. Determine which employees are qualified for a promotion to a manager

Query A	
Business Purpose	*This query lists the names of the managers that have less than 3 employees working for them.* ***The mall requires that all restaurants at the food court have a minimum of 3 employees to ensure that no workers are being overworked.*** *Mission Objective 1*
Data Output Expected	*This query outputs the first and last name of managers that have*

SQL Query Text	`SELECT EMPLOYEE_FNAME, EMPLOYEE_LNAME`
	`FROM T10_EMPLOYEE`
	`WHERE EMPLOYEE_ID IN (`
	`SELECT EMPLOYEE_MANAGER_ID`
	`FROM T10_EMPLOYEE`
	`WHERE EMPLOYEE_MANAGER_ID IS NOT NULL`
	`GROUP BY EMPLOYEE_MANAGER_ID`
	`HAVING COUNT(employee_id) < 3);`
	
	`/* Team 10 12-06-2022 */`

SQL Query Screenshot	

Query B	
Business Purpose	*The purpose of this query is to show which employees are generating a higher level of revenue (between 20-50). The business reason for this range is that $20 of revenue is the target amount for each employee and $50 is the maximum. This range will show which employees are above the target and below the maximum. These employees may qualify for a bonus, but it can also indicate that if an employee is nearing the max they are overworked.* *Mission Objective 2*
Data Output Expected	*This query shows the employee IDs and employee name of employees who have generated between $50 and $20 dollars of revenue by fulfilling orders.*
SQL Query Text	`select o.employee_id, e.employee_fname, e.employee_lname sum(mi.menu_item_price) as 'Total Revenue' from t10_orders as o inner join t10_meal as m on m.order_id = o.order_id inner join t10_menu_item as mi on mi.menu_item_id = m.menu_item_id inner join t10_employee as e on e.employee_id = o.employee_id group by o.employee_id, e.employee_fname, e.employee_lname having sum(mi.menu_item_price) between 20 and 50;`  `/* Team 10 12-06-2022 */`
SQL Query Screenshot	

```
select o.employee_id, e.employee_fname, e.employee_lname, sum(mi.menu_item_price) as 'Total Revenue'
from t10_orders as o
    inner join t10_meal as m on m.order_id = o.order_id
    inner join t10_menu_item as mi on mi.menu_item_id = m.menu_item_id
    inner join t10_employee as e on e.employee_id = o.employee_id
group by o.employee_id, e.employee_fname, e.employee_lname
having sum(mi.menu_item_price) between 20 and 50;

/* Team 12-06-2022 */
```

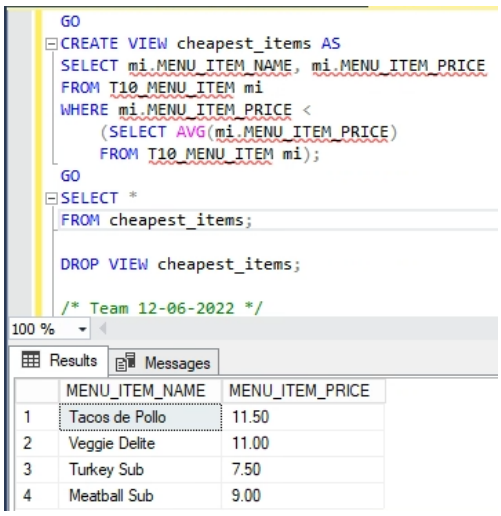
100 %

Results | Messages

	employee_id	employee_fname	employee_lname	Total Revenue
1	15	White	Christina	42.00
2	17	Clark	Maddie	22.00

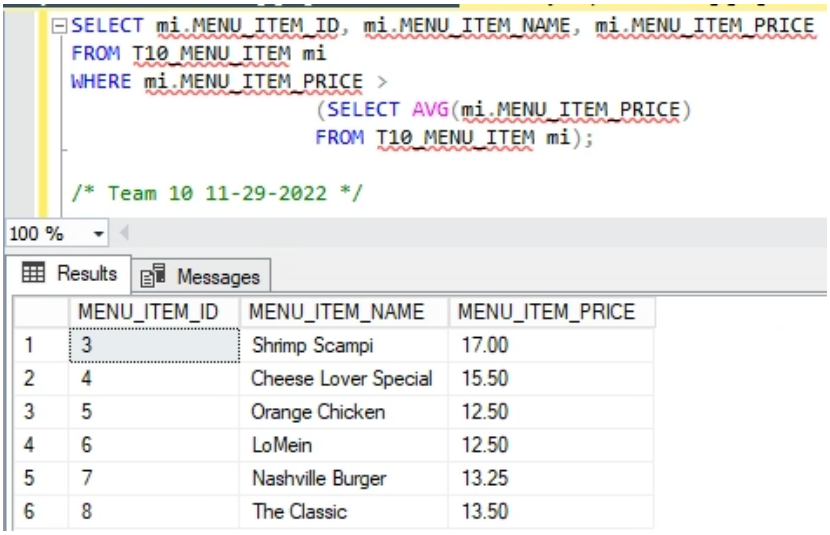
Query C	
Business Purpose	*The goal of this query is to be able to tell the customer which menu items are cheapest. This will allow the end user to determine which items are a bargain.* *Mission Objective 7*
Data Output Expected	*This query outputs the menu item name and menu item price for the menu items that are below the average menu item price*

SQL Query Text	`GO`
	`CREATE VIEW cheapest_items AS`
	`SELECT mi.MENU_ITEM_NAME,`
	`mi.MENU_ITEM_PRICE`
	`FROM T10_MENU_ITEM mi`
	`WHERE mi.MENU_ITEM_PRICE <`
	`(SELECT AVG(mi.MENU_ITEM_PRICE)`
	`FROM T10_MENU_ITEM mi);`
	`GO`
	`SELECT *`
	`FROM cheapest_items;`
	
	`DROP VIEW cheapest_items;`
	
	`/* Team 12-06-2022 */`

SQL Query Screenshot	

Query D	
Business Purpose	*This query allows us to see which menu items are above the average price. Pricing can be an indicator of meal item popularity. It also allows the individual restaurant to compare their prices across all restaurants that have 'premium-priced' items. This query can help restaurants adjust their price as needed to increase competition and gain more orders* *Mission Objective 11*
Data Output Expected	*The output shows a list of the menu items, its name, and its price.*

SQL Query Text	`SELECT mi.MENU_ITEM_ID, mi.MENU_ITEM_NAME, mi.MENU_ITEM_PRICE`
	`FROM T10_MENU_ITEM mi`
	`WHERE mi.MENU_ITEM_PRICE >`
	`(SELECT AVG(mi.MENU_ITEM_PRICE)`
	`FROM T10_MENU_ITEM mi);`
	
	`/* Team 10 11-29-2022 */`

SQL Query Screenshot	

```
SELECT mi.MENU_ITEM_ID, mi.MENU_ITEM_NAME, mi.MENU_ITEM_PRICE
FROM T10_MENU_ITEM mi
WHERE mi.MENU_ITEM_PRICE >
                    (SELECT AVG(mi.MENU_ITEM_PRICE)
                    FROM T10_MENU_ITEM mi);

/* Team 10 11-29-2022 */
```

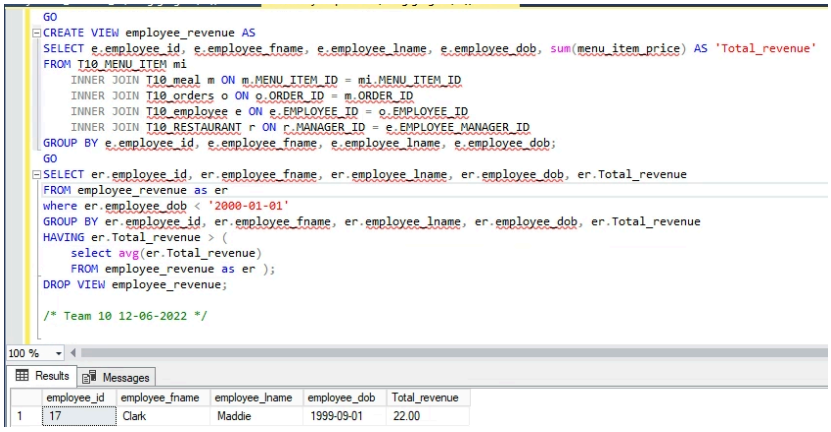
100 %

Results | Messages

	MENU_ITEM_ID	MENU_ITEM_NAME	MENU_ITEM_PRICE
1	3	Shrimp Scampi	17.00
2	4	Cheese Lover Special	15.50
3	5	Orange Chicken	12.50
4	6	LoMein	12.50
5	7	Nashville Burger	13.25
6	8	The Classic	13.50

Query E	
Business Purpose	*This query finds current employees that are qualified to be promoted to a manager. The qualifications include being at least 22 years of age, has generated more than the average amount of revenue, and has not been a manager before.*
Data Output Expected	*This outputs the employee id , the employee first and last name, the employee date of birth, and total revenue generated given than they are at least 22 years old and have generated more than the average amount of revenue*

SQL Query Text	`GO`
	`CREATE VIEW employee_revenue AS`
	`SELECT e.employee_id, e.employee_fname,`
	`e.employee_lname, e.employee_dob,`
	`sum(menu_item_price) AS 'Total_revenue'`
	`FROM T10_MENU_ITEM mi`
	`INNER JOIN T10_meal m ON m.MENU_ITEM_ID`
	`= mi.MENU_ITEM_ID`
	`INNER JOIN T10_orders o ON o.ORDER_ID =`
	`m.ORDER_ID`
	`INNER JOIN T10_employee e ON`
	`e.EMPLOYEE_ID = o.EMPLOYEE_ID`
	`INNER JOIN T10_RESTAURANT r ON`
	`r.MANAGER_ID = e.EMPLOYEE_MANAGER_ID`
	`GROUP BY e.employee_id, e.employee_fname,`
	`e.employee_lname, e.employee_dob;`
	`GO`
	`SELECT er.employee_id, er.employee_fname,`
	`er.employee_lname, er.employee_dob,`
	`er.Total_revenue`
	`FROM employee_revenue as er`
	`where er.employee_dob < '2000-01-01'`
	`GROUP BY er.employee_id, er.employee_fname,`
	`er.employee_lname, er.employee_dob,`
	`er.Total_revenue`
	`HAVING er.Total_revenue > (`
	`select avg(er.Total_revenue)`
	`FROM employee_revenue as er );`
	`DROP VIEW employee_revenue;`

	/* Team 10 12-06-2022 */
SQL Query Screenshot	

```sql
GO
CREATE VIEW employee_revenue AS
SELECT e.employee_id, e.employee_fname, e.employee_lname, e.employee_dob, sum(menu_item_price) AS 'Total_revenue'
FROM T10_MENU_ITEM mi
    INNER JOIN T10_meal m ON m.MENU_ITEM_ID = mi.MENU_ITEM_ID
    INNER JOIN T10_orders o ON o.ORDER_ID = m.ORDER_ID
    INNER JOIN T10_employee e ON e.EMPLOYEE_ID = o.EMPLOYEE_ID
    INNER JOIN T10_RESTAURANT r ON r.MANAGER_ID = e.EMPLOYEE_MANAGER_ID
GROUP BY e.employee_id, e.employee_fname, e.employee_lname, e.employee_dob;
GO
SELECT er.employee_id, er.employee_fname, er.employee_lname, er.employee_dob, er.Total_revenue
FROM employee_revenue as er
where er.employee_dob < '2000-01-01'
GROUP BY er.employee_id, er.employee_fname, er.employee_lname, er.employee_dob, er.Total_revenue
HAVING er.Total_revenue > (
    select avg(er.Total_revenue)
    FROM employee_revenue as er );
DROP VIEW employee_revenue;

/* Team 10 12-06-2022 */
```

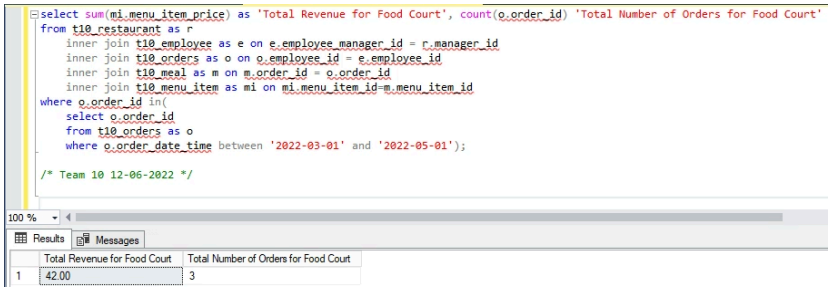
100 %

Results | Messages

	employee_id	employee_fname	employee_lname	employee_dob	Total_revenue
1	17	Clark	Maddie	1999-09-01	22.00

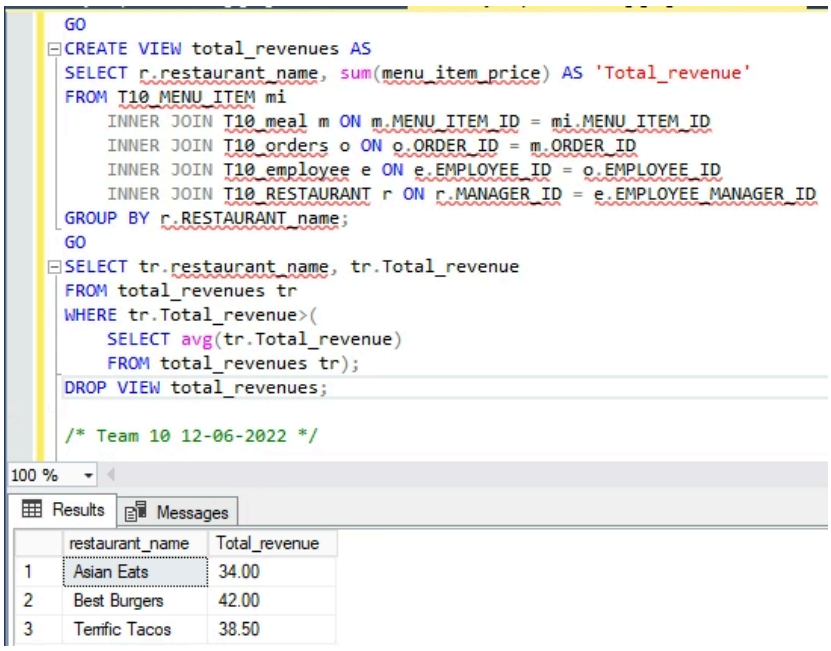
Query F1	
Business Purpose	*This query shows the total revenue and total number of orders between 03/01/2022 and 05/01/2022. This will allow the mall to determine if certain seasons are more popular than others. They can then adjust their marketing efforts depending on the time of year.* *Mission objective 11*
Data Output Expected	*This outputs the total dollars worth of revenue and the total number of orders placed in the Spring months.*

SQL Query Text	`select sum(mi.menu_item_price) as 'Total Revenue for Food Court', count(o.order_id) 'Total Number of Orders for Food Court'`
	`from t10_restaurant as r`
	`inner join t10_employee as e on e.employee_manager_id = r.manager_id`
	`inner join t10_orders as o on o.employee_id = e.employee_id`
	`inner join t10_meal as m on m.order_id = o.order_id`
	`inner join t10_menu_item as mi on mi.menu_item_id=m.menu_item_id`
	`where o.order_id in(`
	`select o.order_id`
	`from t10_orders as o`
	`where o.order_date_time between '2022-03-01' and '2022-05-01');`
	
	`/* Team 10 12-06-2022 */`

SQL Query Screenshot	

Query F2	
Business Purpose	*This query allows us to see the total revenue for restaurants with total revenue than the average total revenues for all the restaurants in the database. It will allow the food court to recognize which restaurant is the most successful* *Mission Objective 6*
Data Output Expected	*The restaurant names and their total revenue for restaurants with total revenue greater than the average total revenues for all the restaurant*

SQL Query Text	`GO`
	`CREATE VIEW total_revenues AS`
	`SELECT r.restaurant_name, sum(menu_item_price) AS 'Total_revenue'`
	`FROM T10_MENU_ITEM mi`
	`INNER JOIN T10_meal m ON m.MENU_ITEM_ID = mi.MENU_ITEM_ID`
	`INNER JOIN T10_orders o ON o.ORDER_ID = m.ORDER_ID`
	`INNER JOIN T10_employee e ON e.EMPLOYEE_ID = o.EMPLOYEE_ID`
	`INNER JOIN T10_RESTAURANT r ON r.MANAGER_ID = e.EMPLOYEE_MANAGER_ID`
	`GROUP BY r.RESTAURANT_name;`
	`GO`
	`SELECT tr.restaurant_name, tr.Total_revenue`
	`FROM total_revenues tr`
	`WHERE tr.Total_revenue>(`
	`SELECT avg(tr.Total_revenue)`
	`FROM total_revenues tr);`
	`DROP VIEW total_revenues;`
	`/* Team 10 12-06-2022 */`

SQL Query Screenshot	

```
GO
CREATE VIEW total_revenues AS
SELECT r.restaurant_name, sum(menu_item_price) AS 'Total_revenue'
FROM T10_MENU_ITEM mi
    INNER JOIN T10_meal m ON m.MENU_ITEM_ID = mi.MENU_ITEM_ID
    INNER JOIN T10_orders o ON o.ORDER_ID = m.ORDER_ID
    INNER JOIN T10_employee e ON e.EMPLOYEE_ID = o.EMPLOYEE_ID
    INNER JOIN T10_RESTAURANT r ON r.MANAGER_ID = e.EMPLOYEE_MANAGER_ID
GROUP BY r.RESTAURANT_name;
GO
SELECT tr.restaurant_name, tr.Total_revenue
FROM total_revenues tr
WHERE tr.Total_revenue>(
    SELECT avg(tr.Total_revenue)
    FROM total_revenues tr);
DROP VIEW total_revenues;

/* Team 10 12-06-2022 */
```

100 %

Results | Messages

	restaurant_name	Total_revenue
1	Asian Eats	34.00
2	Best Burgers	42.00
3	Terrific Tacos	38.50