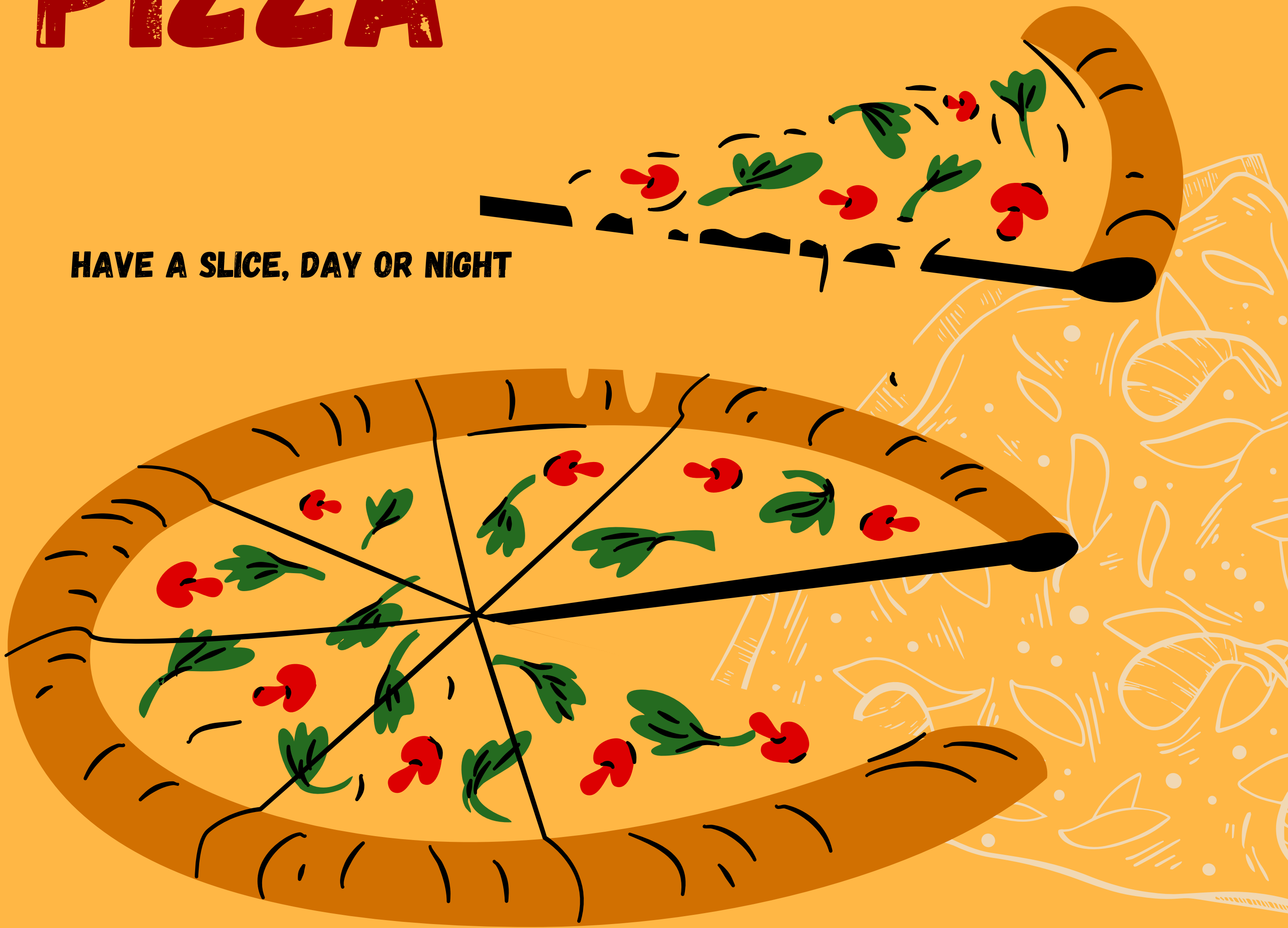


LEKS PIZZA

HAVE A SLICE, DAY OR NIGHT



2020 SALES - SQL ANALYSIS

Github link: <https://github.com/leks39/Leks-Pizza>

Data Import from Excel

```
-- Importing data from Excel

-- Creating Table 1 - delivery info
CREATE TABLE delivery_info(
    delivery_id int, rider_name varchar(50), location_type varchar(50),
    distance_km float, order_time varchar(50), deliverytime_mins int,
    delivery_location varchar(50));

-- Creating Table 2 - order info
CREATE TABLE order_info(
    order_id int, order_channel varchar(50), order_date varchar(50),
    pizza_size varchar(50), pizza_type varchar(50), side_order varchar(50),
    order_amount float, tip float, payment_method varchar(50));

-- Creating Table 3 - pizza info
CREATE TABLE pizza_info (
    pizza_type varchar(50), menu_name varchar(50), category varchar(50),
    ingredients varchar(250));
```

Question 1

-- Calculate Sales, Quantity, Average Order Price for each month in 2020

-- Function used - Table Update, Aliases, Round(), Sub-Queries

Initial step involves converting order_date column to datetime format

Update order_info

Set order_date = str_to_date(order_date, "%m/%d/%Y");

Select *, Round(Month_Sales/Month_Quantity,2) as average_ticket

From (Select monthname(order_date) as month_Name, Round(SUM(order_amount),2)
as Month_Sales, count(order_id) as quantity

From order_info

Group By Month_Name

Order By FIELD(Month_Name, 'January', 'February', 'March', 'April', 'May', 'June',
'July', 'August', 'September', 'October', 'November', 'December'))a;

-- Result

Month_Name	Month_Sales	Month_Quantity	Average_Ticket
January	5952.17	268	22.21
February	6228.4	288	21.63
March	10552.64	482	21.89
April	9822.9	441	22.27
May	13248.22	597	22.19
June	9947.53	459	21.67
July	5299.74	241	21.99
August	9958.29	461	21.6
September	7889.9	350	22.54
October	6602.37	301	21.93
November	10134.58	472	21.47
December	15823.57	623	25.4

Question 2

-- Question 2 - Rank Days of week in terms of sales and quantity sold.

-- Functions used: DateTime, Rank

-- Query

```
Select dayname(order_date) as day_name , round(sum(order_amount),2)
as day_sales, count(order_id) as day_quantity,
rank() over (Order By Sum(order_amount) Desc) as sales_Rank
From order_info
Group By Day_of_Week;
```

-- Result

day_name	day_sales	day_quantity	sales_rank
Sunday	29333.39	1306	1
Saturday	23701.61	1045	2
Friday	21010.03	944	3
Thursday	10661.61	488	4
Tuesday	9860.28	433	5
Wednesday	8992.37	404	6
Monday	7901.03	363	7

Question 3

```
-- Question 3 - Get Total Sales, Total Quantity, Total Tip, Tip Percent by Rider

-- Function used: Generated Columns, Joins

--Query
-- First step is to create a new column for tip percent using generated column function
Alter Table order_info
Add Column tip_percent float Generated Always as ((tip/order_amount)*100);

Select d.rider_name as rider_name , round(sum(o.order_amount),2)
as sales_amount, count(o.order_id) as order_no,
round(avg(d.deliverytime_mins),2) as avg_del_mins,
round(sum(o.tip),2) as total_tips, round(tip_percent,2) as tip_percent
From delivery_info as d
join order_info as o ON delivery_id = order_id
Group by Rider_Name
Order by Sales_Amount Desc;

-- Result
```

rider_name	sales_amount	order_no	avg_del_mins	total_Tips	tip_percent
Angel	37600.75	1709	20.12	5768.36	38.85
Sammie	33326.95	1488	19.69	5089.25	8.7
Tim	21565.08	939	19.89	2981.99	6.96
Paul	18967.55	847	21.09	3132.04	26.13

Question 4

```
-- Divide deliveries into time periods with quantity and percentage total of quantity  
  
-- Functions used: With Clause, Datatype conversion, Case When Statements  
  
--Query  
-- First step is converting the order_time column in delivery_info table
```

```
Update delivery_info
```

```
Set order_time = TIME_FORMAT(order_time, '%H:%i');
```

```
with total as
```

```
    ( select count(delivery_id) as total  
      from delivery_info )
```

```
Select order_period, count(order_period) as period_quantity,  
round(count(order_period)/t.total*100,2) as percent_of_total  
From (Select delivery_id, rider_name,location_type, deliverytime_mins,  
      Case When order_time < '13:00' then 'Early Orders'  
      When order_time > '13:00' and order_time < '17:00' then 'Lunch Orders'  
      When order_time > '17:00' and order_time < '21:00' then 'Dinner Orders'  
      Else 'Late Orders'  
      End as 'order_period'  
From delivery_info) as period, total as t  
Group by order_period  
Order by period_quantity Desc;
```

```
-- Result
```

order_period	period_quantity	percent_of_total
Dinner Orders	2303	46.21
Lunch Orders	1810	36.32
Early Orders	805	16.15
Late Orders	66	1.32

Question 5

-- Top 10 selling pizzas and their categories

-- Functions used: Limit

--Query

```
Select o.pizza_type, p.category, count(o.pizza_type) as quantity_sold
From order_info as o
Join pizza_info as p
ON o.pizza_type = p.pizza_type
Group by pizza_type
Order by quantity_sold desc
Limit 10;
```

-- Result

pizza_type	category	quantity_sold
ital_supr	Meat	433
pepperoni	Classic	304
bbq_ckn	Chicken	247
sicilian	Meat	238
thai_ckn	Chicken	234
cali_ckn	Chicken	230
classic_dlx	Classic	228
hawaiian	Classic	218
spicy_ital	Meat	185
southw_ckn	Chicken	183

Question 6

```
-- Top 3 selling pizzas in each category categories

-- Functions used: Row Number, With Clause

--Query
with pizza_sales as (Select o.pizza_type, p.menu_name,
p.category, count(o.pizza_type) as qty_sold,
row_number() over (partition by category order by count(o.pizza_type) desc)
as category_rank from order_info as o
join pizza_info as p on o.pizza_type = p.pizza_type
Group by pizza_type, category
order by category)

Select *
From pizza_sales
Where category_rank ≤ 3;
```

-- Result

pizza_type	menu_name	category	qty_sold	cat_rank
bbq_ckn	Barbecue Chicken	Chicken	247	1
thai_ckn	Thai Chicken	Chicken	234	2
cali_ckn	California Chicken	Chicken	230	3
pepperoni	Pepperoni Pizza	Classic	304	1
classic_dlx	Classic Deluxe	Classic	228	2
hawaiian	Hawaiian Pizza	Classic	218	3
ital_supr	Italian Supreme	Meat	433	1
sicilian	Sicilian Pizza	Meat	238	2
spicy_ital	Spicy Italian	Meat	185	3
four_cheese	Four Cheese	Veggie	173	1
veggie_veg	Vegetables Pizza	Veggie	171	2
five_cheese	The Five Cheese	Veggie	149	3

Question 7

-- Location details

--Query

```
Select d.delivery_location, round(sum(o.order_amount),2) as zone_sales,
count(o.order_id) as zone_orders,round(avg(d.distance_km),2) as avg_dist_km,
round(avg(d.deliverytime_mins),2) as avg_del_min
From delivery_info as d
Join order_info as o ON d.delivery_id = o.order_id
Group by delivery_location
Order by location_sales desc;
```

-- Result

delivery_location	zone_sales	zone_orders	avg_dist_km	avg_del_min
White Water	22917.05	1057	1.79	15.43
Collister North	20367.67	911	1.31	10.97
Garden City	19307.69	833	2.54	22.41
NorthEnd	15857.52	752	2.89	25.76
Collister South	11706.65	545	1.17	9.71
Bogus Basin	9780.58	412	3.23	29.34
West Glennwood	8266.86	357	3.02	26.83
HiddenSprings	3256.31	116	7.71	76.89

Question 8

```
-- Divide deliveries into speed brackets based on deliverytime_mins
-- Calculate percentage of deliveries that are late

-- Functions used: With Clause, Sub-Queries

--Query
with total_deliveries as
(select count(delivery_id) as total_orders from delivery_info)

Select delivery_speed, round(delivery_count/td.total_orders*100,2) as del_percent
From(
Select delivery_speed, count(delivery_speed) as delivery_count
From (
Select *,
Case when deliverytime_mins <= 10 then 'Fast Delivery'
When deliverytime_mins > 10 and deliverytime_mins <=30 then 'Standard Delivery'
Else 'Late Delivery'
End as 'delivery_speed'
From delivery_info)ds
Group by delivery_speed
Order by delivery_count desc)dc, total_deliveries as td
Having delivery_speed = 'Late Delivery';

-- Result 1
delivery_speed    delivery_count
Fast Delivery     1263
Standard Delivery 3081
Late Delivery     640

-- Result 2
delivery_speed    delivery_percent
Late Delivery     12.84
```

Question 9

-- Calculate the number of fast deliveries per Rider

--Query

```
Select rider_name, count(delivery_speed) as no_of_deliveries
From(
Select *,
Case when deliverytime_mins < 10 then 'Fast Delivery'
When deliverytime_mins > 10 and deliverytime_mins < 30 then 'Standard Delivery'
Else 'Late Delivery'
End as 'delivery_speed'
From delivery_info)ds
Where delivery_speed = 'Fast Delivery'
group by delivery_speed, rider_name
Order by no_of_deliveries Desc;
```

-- Result

rider_name	no_of_deliveries
Sammie	382
Angel	372
Tim	199
Paul	176

Question 10

```
-- Calculate number of deliveries 30% above the average order amount
-- by location

-- Function used: Sub-Queries, With Clause, Joins

--Query
with big_order as (
  Select *
  From order_info
  Where order_amount >
    (Select round(avg(order_amount),2) * 1.3
     From order_info))

Select d.delivery_location, count(bo.order_id) as big_orders
From big_order as bo
Join delivery_info as d
on bo.order_id = d.delivery_id
Group by d.delivery_location
Order by big_orders desc;

-- Result
delivery_location    big_orders
Garden City          161
Collister North      156
White Water          147
Bogus Basin          87
NorthEnd              84
Collister South       79
West Glennwood        75
HiddenSprings         24
```