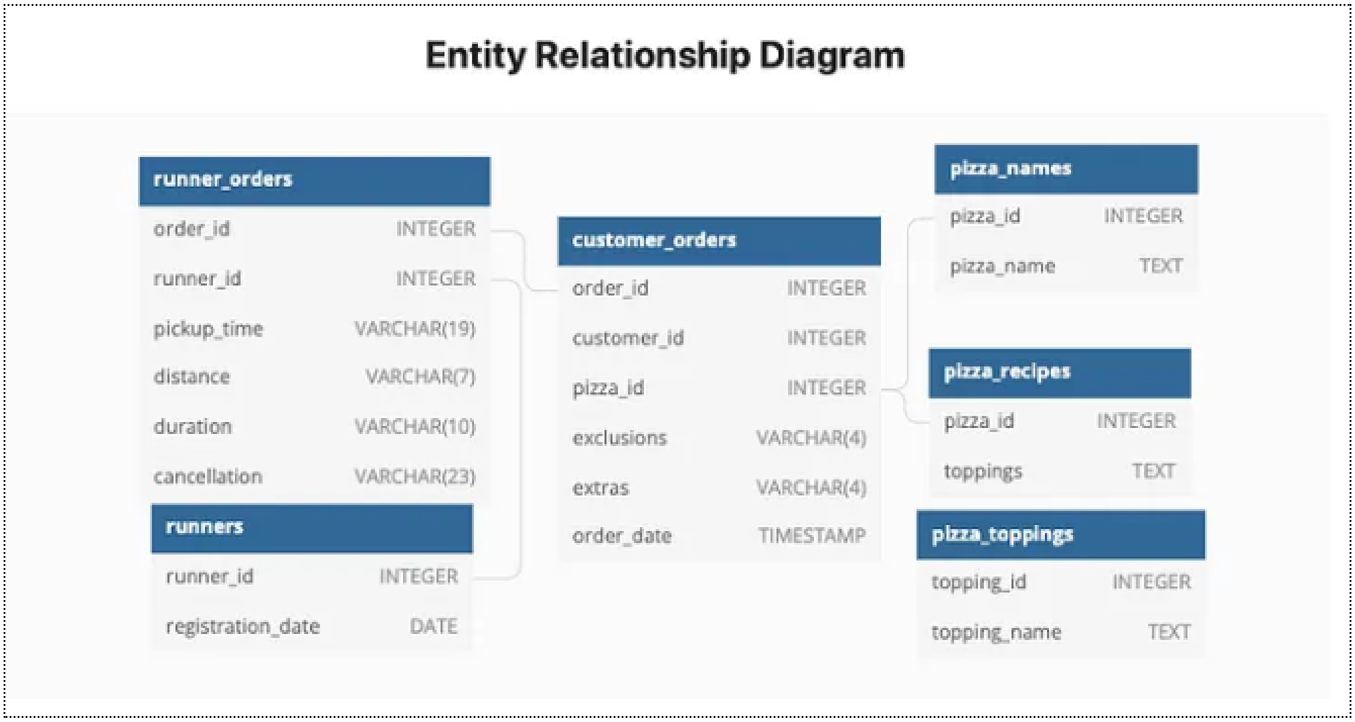
Case Study #2 - Pizza Runner

In this challenge we have 6 tables — runner_orders, runners, customer_orders, pizza_names, pizza_recipes, pizza_toppings (see ERD below):



Entity Relationship Diagram

runner_orders	
order_id	INTEGER
runner_id	INTEGER
pickup_time	VARCHAR(19)
distance	VARCHAR(7)
duration	VARCHAR(10)
cancellation	VARCHAR(23)

runners	
runner_id	INTEGER
registration_date	DATE

customer_orders	
order_id	INTEGER
customer_id	INTEGER
pizza_id	INTEGER
exclusions	VARCHAR(4)
extras	VARCHAR(4)
order_date	TIMESTAMP

pizza_names	
pizza_id	INTEGER
pizza_name	TEXT

pizza_recipes	
pizza_id	INTEGER
toppings	TEXT

pizza_toppings	
topping_id	INTEGER
topping_name	TEXT

User requires to clean his data and apply some basic calculations so he can better direct his runners and optimize Pizza Runner's operations.

This case study has lots of questions - they are broken up by area of focus including:

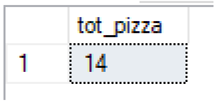
- Pizza Metrics
- Runner and Customer Experience
- Ingredient Optimisation
- Pricing and Ratings

Case Study Questions

```
select * from dbo.customer_orders;
select * from dbo.pizza_names;
select * from dbo.pizza_recipes;
select * from dbo.pizza_toppings;
select * from dbo.runner_orders;
select * from dbo.runners;
```

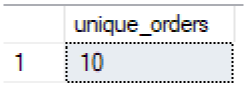
A. Pizza Metrics

```
– 1. How many pizzas were ordered ?

select count (pizza_id) as tot_pizza from dbo.customer_orders
```



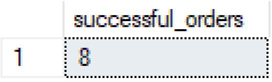
	tot_pizza
1	14

```
-- 2. How many unique customer orders were made ?

select count(distinct order_id) as unique_orders from dbo.customer_orders
```



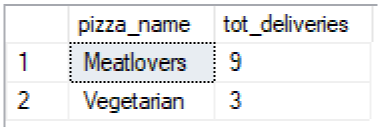
	unique_orders
1	10

```
-- 3. How many successful orders were delivered by each runner?

select count(distinct order_id) as [successful_orders] from dbo.customer_orders where order_id not in (
select order_id from dbo.runner_orders where cancellation like '%cancel%')
```



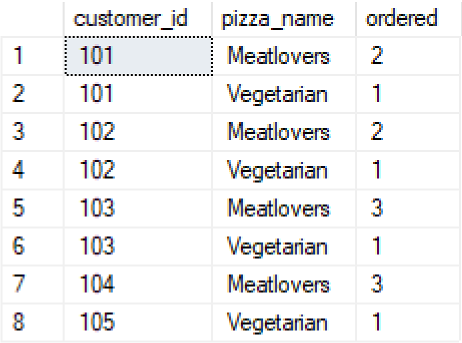
	successful_orders
1	8

```
-- 4. How many of each type of pizza was delivered?

select pizza_name, tot_deliveries from (
select pizza_id, count(pizza_id) as tot_deliveries  from dbo.customer_orders where order_id not in
(select order_id from dbo.runner_orders where cancellation like '%cancel%') group by pizza_id)x join
dbo.pizza_names p on p.pizza_id=x.pizza_id
```



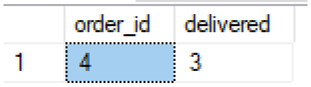
	pizza_name	tot_deliveries
1	Meatlovers	9
2	Vegetarian	3

```
-- 5. How many Vegetarian and Meatlovers were ordered by each customer?

select customer_id, pizza_name, ordered from (
select customer_id, pizza_id, count(pizza_id) as ordered from dbo.customer_orders group by pizza_id,
customer_id)x join dbo.pizza_names p
on p.pizza_id=x.pizza_id order by customer_id
```



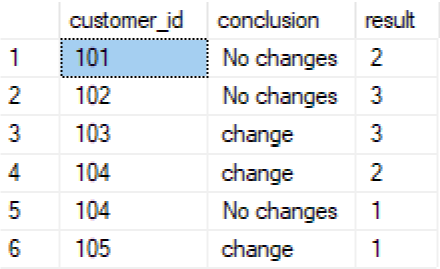
	customer_id	pizza_name	ordered
1	101	Meatlovers	2
2	101	Vegetarian	1
3	102	Meatlovers	2
4	102	Vegetarian	1
5	103	Meatlovers	3
6	103	Vegetarian	1
7	104	Meatlovers	3
8	105	Vegetarian	1

```sql
-- 6. What was the maximum number of pizzas delivered in a single order?

select top 1 order_id, count(pizza_id) as delivered from dbo.customer_orders where order_id not in
(select order_id from dbo.runner_orders where cancellation like '%cancel%') group by order_id
order by delivered desc
```



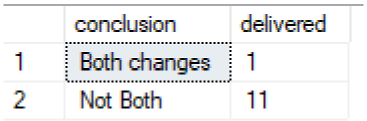
	order_id	delivered
1	4	3

```sql
-- 7. For each customer, how many delivered pizzas had at least 1 change and how many had no changes?

select customer_id, conclusion, count(conclusion) as result from (
select customer_id, pizza_id, exclude, extra, case when exclude='0' and extra='0' then 'No changes' else
'change' end as conclusion from (
select *,
case when exclusions is null or exclusions like 'null%' or exclusions=' ' then '0' else exclusions end as exclude,
case when extras='NULL' or extras is null or extras=' ' then '0' else extras end as extra
from dbo.customer_orders where order_id not in
(select order_id from dbo.runner_orders where cancellation like '%cancel%'))x)y group by customer_id,
conclusion order by customer_id
```



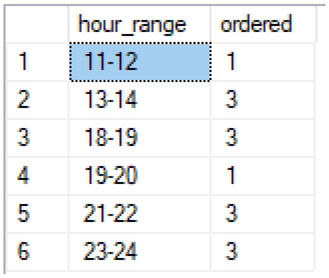
	customer_id	conclusion	result
1	101	No changes	2
2	102	No changes	3
3	103	change	3
4	104	change	2
5	104	No changes	1
6	105	change	1

```sql
-- 8. How many pizzas were delivered that had both exclusions and extras?

select conclusion, count(pizza_id) as delivered from (
select customer_id, pizza_id, exclude, extra, case when exclude<>'0' and extra<>'0' then 'Both changes' else
'Not Both' end as conclusion from (
select *,
case when exclusions is null or exclusions like 'null%' or exclusions=' ' then '0' else exclusions end as exclude,
case when extras='NULL' or extras is null or extras=' ' then '0' else extras end as extra
from dbo.customer_orders where order_id not in
(select order_id from dbo.runner_orders where cancellation like '%cancel%'))x)y group by conclusion
```



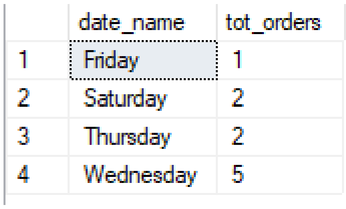
	conclusion	delivered
1	Both changes	1
2	Not Both	11

```sql
-- 9. What was the total volume of pizzas ordered for each hour of the day?

select hour_range, count(pizza_id) as ordered from (
select *, concat(DATEPART(hour, order_time),'-',DATEPART(hour,order_time)+1) as hour_range from
dbo.customer_orders)x group by hour_range
```



	hour_range	ordered
1	11-12	1
2	13-14	3
3	18-19	3
4	19-20	1
5	21-22	3
6	23-24	3

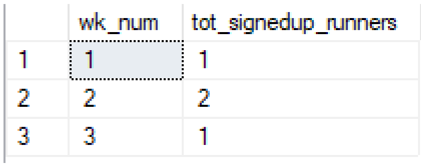
```
-- 10. What was the volume of orders for each day of the week?

select date_name, count(distinct order_id) as tot_orders from (
select *, datename(dw,order_time) as date_name from dbo.customer_orders)x group by date_name
```



	date_name	tot_orders
1	Friday	1
2	Saturday	2
3	Thursday	2
4	Wednesday	5

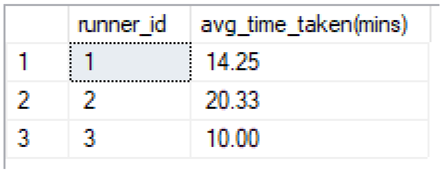
B. Runner and Customer Experience

```
-- 1. How many runners signed up for each 1 week period? (i.e. week starts 2021-01-01)

select wk_num, count(runner_id) as tot_signedup_runners from (
select *, DATEPART(week,registration_date) as wk_num from dbo.runners where
registration_date>='2021-01-01')x group by wk_num
```



	wk_num	tot_signedup_runners
1	1	1
2	2	2
3	3	1

```
-- 2. What was the average time in minutes it took for each runner to arrive at the Pizza Runner HQ to pickup
the order?
 time to arrive = pickup_time - order_time

select runner_id, cast(tot_time*1.0/tot_orders as dec(4,2)) as [avg_time_taken(mins)] from
(select runner_id, count(order_id) as tot_orders, sum(time_taken) as tot_time from (
select runner_id, order_id, max(time_taken) as time_taken from(
select c.order_id, runner_id,order_time, pickup, DATEDIFF(minute, order_time, pickup) time_taken from (
select *, convert(datetime2,pickup_time) as pickup from dbo.runner_orders where pickup_time <> 'null')x
join dbo.customer_orders c on c.order_id=x.order_id)y group by runner_id, order_id)z group by runner_id)q
```

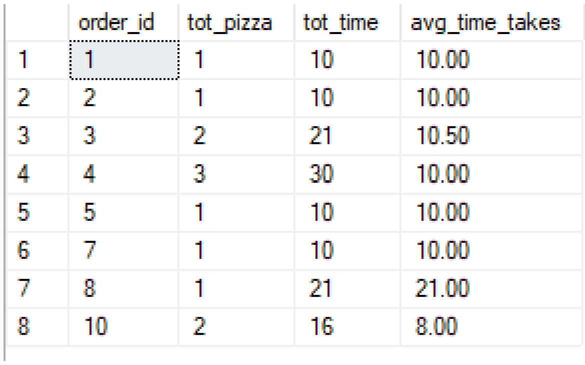


	runner_id	avg_time_taken(mins)
1	1	14.25
2	2	20.33
3	3	10.00

```
-- 3. Is there any relationship between the number of pizzas and how long the order takes to prepare?
```

```
select *, cast(tot_time*1.0/tot_pizza as dec(4,2)) as avg_time_takes from (
select order_id, count(pizza_id) tot_pizza, max(time_taken) as tot_time from
(select c.order_id, runner_id,c.pizza_id,order_time, pickup, DATEDIFF(minute, order_time, pickup) time_taken
from (
select *, convert(datetime,pickup_time) as pickup from dbo.runner_orders where pickup_time <> 'null')x
join dbo.customer_orders c on c.order_id=x.order_id)y group by y.order_id)z
```

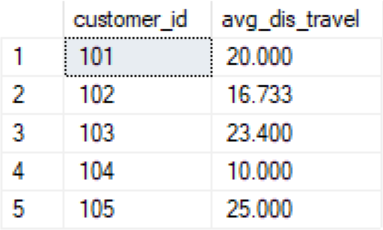
-- we find that all order_id from 1 to 7 takes 10 mins time to prepare 1 pizza, but Order_Id 8 took 21 mins whereas order_id 10 took 8 mins, which varies from the above pattern. If we exclude Order_id(8,10) we can say that there exists a relationship between the number of pizzas and preparation time.



	order_id	tot_pizza	tot_time	avg_time_takes
1	1	1	10	10.00
2	2	1	10	10.00
3	3	2	21	10.50
4	4	3	30	10.00
5	5	1	10	10.00
6	7	1	10	10.00
7	8	1	21	21.00
8	10	2	16	8.00

-- 4. What was the average distance travelled for each customer?

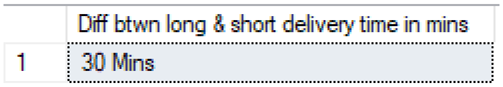
```
select customer_id, convert(dec(8,3),tot_dis/tot_orders) as avg_dis_travel from (
select customer_id, count(customer_id) tot_orders, sum(cast(dist as dec(5,2))) as tot_dis from(
select c.order_id,customer_id, runner_id, distance,
case when right(trim(distance),2)='km' then trim(left(distance,CHARINDEX('k',distance)-1)) else trim(distance)
end as dist from dbo.customer_orders c join dbo.runner_orders r on r.order_id=c.order_id where
distance<>'null')x group by customer_id)y
```



	customer_id	avg_dis_travel
1	101	20.000
2	102	16.733
3	103	23.400
4	104	10.000
5	105	25.000

-- 5. What was the difference between the longest and shortest delivery times for all orders?

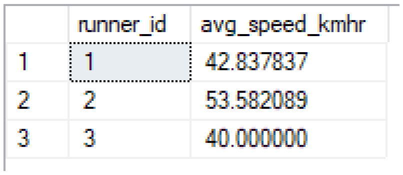
```
select concat(max(dur)-min(dur),' Mins') as [Diff btwn long & short delivery time in mins] from (
select order_id, runner_id, duration,
cast(case when duration like '%min%' then trim(left(duration,CHARINDEX('m',duration)-1)) else trim(duration)
end as int) as dur
from dbo.runner_orders where duration <> 'null')x
```



	Diff btwn long & short delivery time in mins
1	30 Mins

-- 6. What was the average speed for each runner for each delivery and do you notice any trend for these values?

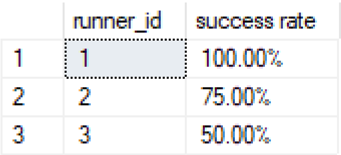
```
select runner_id, sum(dist)/sum(dur_hr) as avg_speed_kmhr from(
select order_id, runner_id, dist, convert(dec(4,2),dur*1.0/60) as dur_hr from(
select order_id, runner_id, distance, duration,
cast(case when right(trim(distance),2)='km' then left(distance,CHARINDEX('k',trim(distance))-1) else
trim(distance) end as dec(5,1)) as dist,
cast(case when duration like '%min%' then trim(left(duration,CHARINDEX('m',duration)-1)) else trim(duration)
end as int) as dur
from dbo.runner_orders where distance<>'null' and duration<>'null')x)y group by runner_id
```



	runner_id	avg_speed_kmhr
1	1	42.837837
2	2	53.582089
3	3	40.000000

-- 7. What is the successful delivery percentage for each runner?

```
select runner_id, concat(cast(status*100.0/tot_orders as dec(5,2)),'%') as [success rate] from
(select runner_id, count(runner_id) as tot_orders, count(case when Delivery='Pass' then 1 else null end) as
status from (
select order_id,runner_id, case when pickup_time ='null' then 'Fail' else 'Pass' end as Delivery from
dbo.runner_orders)x
group by runner_id)y
```

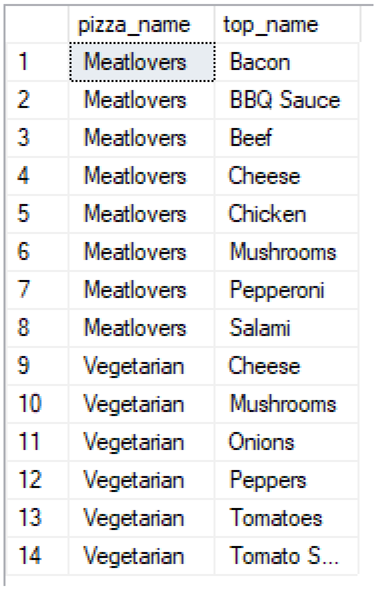


	runner_id	success rate
1	1	100.00%
2	2	75.00%
3	3	50.00%

C. Ingredient Optimisation

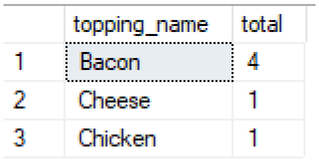
-- 1. What are the standard ingredients for each pizza?

```
select pizza_name, top_name from (
select pizza_id, toppings, cast(topping_name as varchar(255)) as top_name from(
select pizza_id, trim(toppings) as toppings from (
select pizza_id, value as toppings from (
select pizza_id, cast(toppings as varchar(50)) as toppings from dbo.pizza_recipes)x cross apply
string_split(x.toppings,','))k)y
join dbo.pizza_toppings t on t.topping_id=y.toppings)z join dbo.pizza_names p on p.pizza_id=z.pizza_id
```



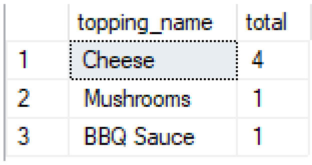
	pizza_name	top_name
1	Meatlovers	Bacon
2	Meatlovers	BBQ Sauce
3	Meatlovers	Beef
4	Meatlovers	Cheese
5	Meatlovers	Chicken
6	Meatlovers	Mushrooms
7	Meatlovers	Pepperoni
8	Meatlovers	Salami
9	Vegetarian	Cheese
10	Vegetarian	Mushrooms
11	Vegetarian	Onions
12	Vegetarian	Peppers
13	Vegetarian	Tomatoes
14	Vegetarian	Tomato S...

```
-- 2. What was the most commonly added extra?

select topping_name, total from(
select extra, count(extra) as total from (
select order_id, trim(value) as extra from (
select order_id, convert(varchar(100),extra) as extra from (
select order_id, extras, case when extras='NULL' or extras=' ' or extras is null then 'N/A' else extras end as
extra from dbo.customer_orders)x
where extra<>'N/A')y cross apply string_split(extra,','))z group by extra)k join dbo.pizza_toppings pt on
pt.topping_id=k.extra
```



	topping_name	total
1	Bacon	4
2	Cheese	1
3	Chicken	1

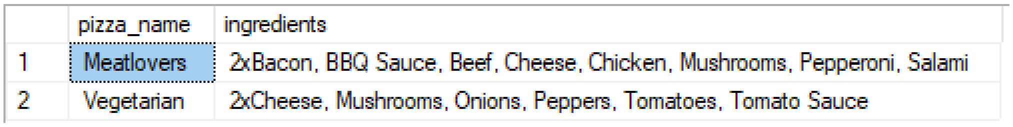
```
-- 3. What was the most common exclusion?

select topping_name, total from(
select exclude, count(exclude) as total from (
select order_id, trim(value) as exclude from (
select order_id, convert(varchar(100),exclude) as exclude from (
select order_id, exclusions, case when exclusions='null' or exclusions=' ' then 'N/A' else exclusions end as
exclude from dbo.customer_orders)x
where exclude<>'N/A')y cross apply string_split(exclude,','))z group by exclude)k join dbo.pizza_toppings pt on
pt.topping_id=k.exclude
order by total desc
```



	topping_name	total
1	Cheese	4
2	Mushrooms	1
3	BBQ Sauce	1

4. Generate an alphabetically ordered comma separated ingredient list for each pizza order from the customer_orders table and add a 2x in front of any relevant ingredients. For example: "Meat Lovers: 2xBacon, Beef, ... , Salami"

```
select pizza_name, concat('2x',ingredients) as ingredients from (
select pizza_id, STRING_AGG(top_name,', ') as ingredients from (
select pizza_id, toppings, cast(topping_name as varchar(255)) as top_name from(
select pizza_id, trim(toppings) as toppings from (
select pizza_id, value as toppings from (
select pizza_id, cast(toppings as varchar(50)) as toppings from dbo.pizza_recipes)x cross apply
string_split(x.toppings,','))k)y
join dbo.pizza_toppings t on t.topping_id=y.toppings)z group by z.pizza_id)q join dbo.pizza_names p on
p.pizza_id=q.pizza_id
```

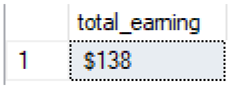


	pizza_name	ingredients
1	Meatlovers	2xBacon, BBQ Sauce, Beef, Cheese, Chicken, Mushrooms, Pepperoni, Salami
2	Vegetarian	2xCheese, Mushrooms, Onions, Peppers, Tomatoes, Tomato Sauce

D. Pricing and Ratings

– 1. A Meat Lovers pizza costs $12 and Vegetarian costs $10 and there were no charges for changes.
How much money has Pizza Runner made so far if there are no delivery fees?

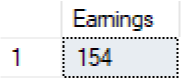
```
select concat('$',sum(price)) as total_earning from (
select order_id, x.pizza_id, pizza_name, case when x.pizza_id=1 then 12 else 10 end as price
from(select * from dbo.customer_orders where order_id not in (select order_id from dbo.runner_orders where
cancellation like '%cancel%'))x
join dbo.pizza_names p on p.pizza_id=x.pizza_id)y
```



	total_earning
1	$138

-- 2. What if there was an additional $1 charge for any pizza extras? Add cheese is $1 extra.

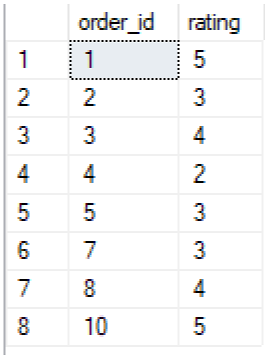
```
select sum(earn) as Earnings from (
select price+extra_cost as earn from (
select *, case when include='N/A' then 0 else 1 end as extra_cost from (
select order_id, pizza_name, price, value as include from (
select order_id, c.pizza_id, pizza_name,case when c.pizza_id=1 then 12 else 10 end as price,extras,
cast(case when extras=' ' or extras is null or extras='null' then 'N/A' else extras end as varchar(100)) as extra
from dbo.customer_orders c join dbo.pizza_names p on p.pizza_id=c.pizza_id
where c.order_id not in(select order_id from dbo.runner_orders where cancellation like '%cancel%'))x cross
apply string_split(extra,','))y)z)k
```



	Earnings
1	154

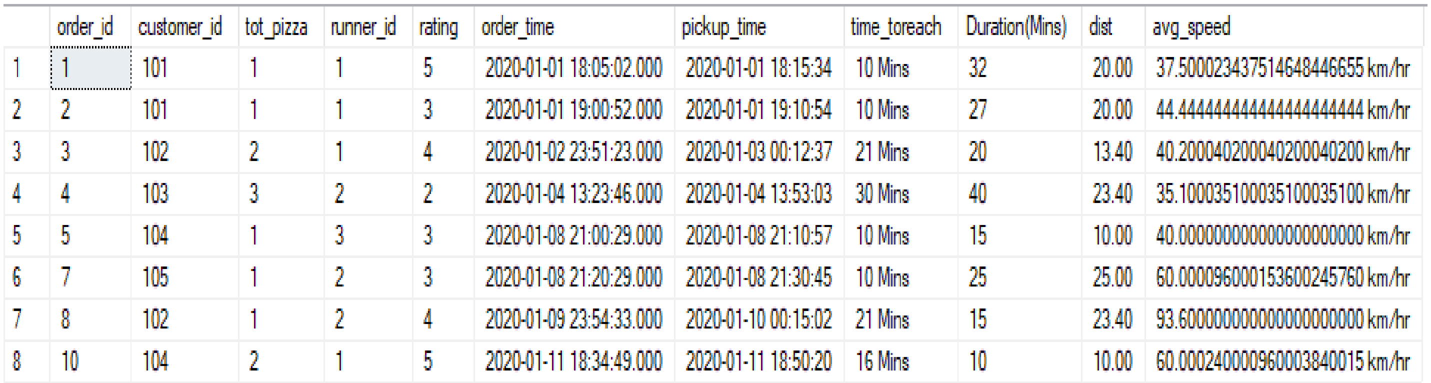
-- 3. The Pizza Runner team now wants to add an additional ratings system that allows customers to rate their runner. How would you design an additional table for this new dataset - generate a schema for this new table and insert your own data for ratings for each successful customer order between 1 to 5.

```
DROP TABLE IF EXISTS ratings;
CREATE TABLE ratings (
order_id int,
rating int);

INSERT INTO ratings VALUES
(1, 5), (2, 3), (3, 4), (4, 2), (5,3), (7, 3), (8, 4), (10, 5);

Select * from ratings;
```



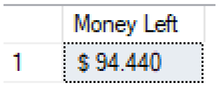
	order_id	rating
1	1	5
2	2	3
3	3	4
4	4	2
5	5	3
6	7	3
7	8	4
8	10	5

```
-- 4. Using your newly generated table - can you join all of the information together to form a table which has
the following information for successful deliveries?
customer_id, order_id, runner_id, rating, order_time, pickup_time, Time between order and pickup, Delivery
duration, Average speed, Total number of pizzas

select *, concat(dist/([Duration(Mins)]*1.0/60),' km/hr') as avg_speed from
(select order_id, customer_id, tot_pizza,runner_id, rating, order_time, pickup_time,concat(DATEDIFF(minute,
order_time,pickup_time),' Mins') as time_toreach,
cast(case when duration like '%min%' then trim(left(duration,CHARINDEX('m',duration)-1)) else trim(duration)
end as int)as [Duration(Mins)],
cast(case when right(trim(distance),2)='km' then trim(left(distance,CHARINDEX('k',distance)-1)) else
trim(distance) end as dec(5,2))as dist from(
select b.order_id,customer_id,tot_pizza,order_time,runner_id,rating,pickup_time,duration,distance from(
select order_id, customer_id, count(pizza_id) as tot_pizza, max(order_time) order_time from (
select * from dbo.customer_orders where order_id not in (select order_id from dbo.runner_orders where
cancellation like '%cancel%'))a
group by order_id, customer_id )b join dbo.runner_orders r on r.order_id=b.order_id join ratings rt on
rt.order_id=b.order_id)c)d
```



	order_id	customer_id	tot_pizza	runner_id	rating	order_time	pickup_time	time_toreach	Duration(Mins)	dist	avg_speed
1	1	101	1	1	5	2020-01-01 18:05:02.000	2020-01-01 18:15:34	10 Mins	32	20.00	37.500023437514648446655 km/hr
2	2	101	1	1	3	2020-01-01 19:00:52.000	2020-01-01 19:10:54	10 Mins	27	20.00	44.444444444444444444444 km/hr
3	3	102	2	1	4	2020-01-02 23:51:23.000	2020-01-03 00:12:37	21 Mins	20	13.40	40.200040200040200040200 km/hr
4	4	103	3	2	2	2020-01-04 13:23:46.000	2020-01-04 13:53:03	30 Mins	40	23.40	35.100035100035100035100 km/hr
5	5	104	1	3	3	2020-01-08 21:00:29.000	2020-01-08 21:10:57	10 Mins	15	10.00	40.000000000000000000000 km/hr
6	7	105	1	2	3	2020-01-08 21:20:29.000	2020-01-08 21:30:45	10 Mins	25	25.00	60.000096000153600245760 km/hr
7	8	102	1	2	4	2020-01-09 23:54:33.000	2020-01-10 00:15:02	21 Mins	15	23.40	93.600000000000000000000 km/hr
8	10	104	2	1	5	2020-01-11 18:34:49.000	2020-01-11 18:50:20	16 Mins	10	10.00	60.000240000960003840015 km/hr

```
-- 5. If a Meat Lovers pizza was $12 and Vegetarian $10 fixed prices with no cost for extras and each runner is
paid $0.30 per kilometer traveled, how much money does Pizza Runner have left over after these deliveries?

select concat('$ ',sum(pizza_earn)-sum(travel_cost)) as [Money Left] from(
select runner_id, sum(tot_price) as pizza_earn, sum(cost_per_km) as travel_cost from(
select *, tot_km*0.3 as cost_per_km from(
select order_id, runner_id, sum(price) as tot_price, max(dist) as tot_km from(
select c.order_id, c.pizza_id,pizza_name, runner_id, case when c.pizza_id=1 then 12 else 10 end as price,
cast(case when right(distance,2)='km' then left(distance,CHARINDEX('k',trim(distance))-1) else trim(distance)
end as dec(5,2)) as dist
from dbo.customer_orders c join dbo.runner_orders r on r.order_id=c.order_id join dbo.pizza_names p on
p.pizza_id=c.pizza_id
where c.order_id not in (select order_id from dbo.runner_orders where cancellation like '%cancel%'))x group
by order_id, runner_id)y)z
group by runner_id)k
```



	Money Left
1	$ 94.440