





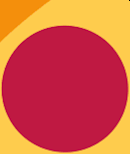
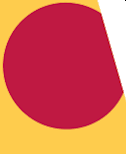
Did you know that over 115 million kilograms of pizza are consumed daily worldwide??? (Well according to Wikipedio onyway...)

Danny was scrolling through his Instagram feed when something really caught his eye - “80s Retro Styling and Pizza Is The Future!”

Danny was sole! on the ic!ea, but he knew that pizza alone was not going to help him get seec! lunching to expanc! his new Pizza Empire - so he hoc! one more genius ic!ea to combine with it - he was going to Uberize it - one! so Pizza Runner was launches!!

Danny started by recruiting “runners” to deliver fresh pizza from Pizza Runner Headquarters (otherwise known as Danny's kouse) and also maxed out his credit card to pay freelance developers to build a mobile app to accept orders from customers.





Danny, a data scientist with a few years of experience, knew that data collection was essential for his business's growth. He created an entity relationship diagram (ERD) for his database design, but he needed help cleaning the data and applying basic calculations. This would allow him to better direct his runners and optimize Pizza Runner's operations.

I am Arckis Rudro, a data analyst. I will use my SQL skills to help Danny make data-driven decisions. I will answer his questions and help him clean and analyze his data. This will allow Danny to make better decisions about how to run his business.

Entity Relationship Diagram

runner orders

pi z z a n am es

customer orders

p1zza rec i pes

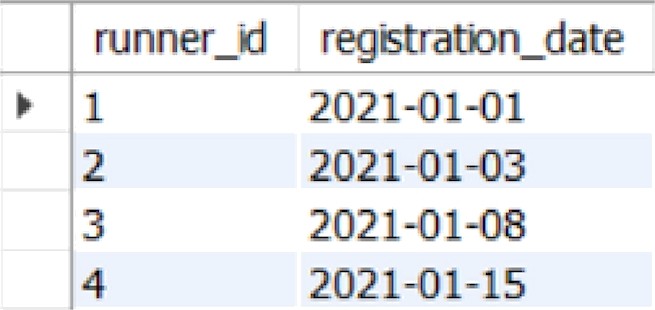
runners





runners:

The runners table shows the registration date for each new runner.

|  |  |
| --- | --- |
| runner\_d | reqstmton\_date |
| y 1 | 2021-01-01 |
| 2 | 2021-01-03 |
| 3 | 2021-01-08 |
| 4 | 2021-01-15 |

customer\_orders:

Customer pizza orders are captured in the customer\_orders fable with 1 row for each individual pizza that is part of the order.

The pizzo\_id relates to the type of pizza which was ordered whilst the exclusions are the ingredient\_id values tkat should be removed from the pizza and the extras are the ingredient\_id values that need to be added to the pizza.

hJote that customers can order multiple pizzas in a single order with varying exclusions and extras values even if the pizza is the same type!





|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| > 1 | 101 | 1 |  |  | 2020-01-01 18:05:02 |
| z | 1 | 1 |  |  | 2O20-0i-0› i9:0»:S2 |
| 3 | 102 | 1 |  |  | 2020-01-02 23:51:Z3 |
| 3 | 102 | 2 |  |  | 2020-01-02 23:51:23 |
| 4 | 103 | 1 | 4 |  | 20Z0-01-04 13:23:46 |
| 4 | 103 | 1 | 4 |  | 2020-01-04 13:23:46 |
| 4 | 103 | 2 | 4 |  | 2020-01-04 13:23:46 |
| 5 | 104 | 1 | rxM | 1 | 2820•01-0821:00:29 |
| 6 | 101 | 2 | rnil | Mž | 2020-01-0821:03:13 |
| 7 | 1DS | 2 | rc5 | 1 | 202O•O1-DB2I:20:29 |
| B | 1D2 | 1 | raJI | rx5 | 2O20-01-0923:54:33 |
| 9 | lDD | 1 | 4 | 1,5 | 29]{EO}-\*O II:2Z:59 |
| 10 | 1Q4 | 1 | riM | nuf | 2020@111 1g34:49 |
| 10 | 104 | 1 | 2, 6 | 1, 4 | 2O20KR1I IB:34:49 |

runners\_orders:

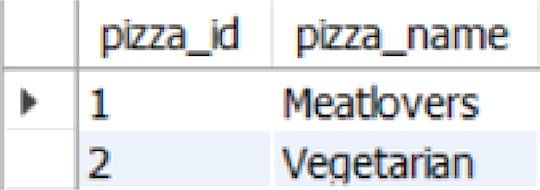
After each order are received througk the system - they are assigned to a runner - however not all orders are fully completed and can be canceled by the restaurant or the customer.

The pickup\_time is the timestamp at which the runner arrives at the Pizza Runner headquarters to pick up the freskly cooked pizzas. The distance and duration fields are related to how far and long the runner had to travel to deliver the order to the respective customer.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| order\_g | runner\_d | picXup\_tme | dBta›ce | durattxi | caxzšatx›o |
| \* 1 | 1 | 2020-01-01 18:15:34 | 20Xm | 32 mčxites |  |
| 2 | 1 | 2020-01-01 19:10:54 | 20Xm | 27 m¥¥Jtes |  |
| 3 | 1 | 2020-01-03 iXl:12:37 | 13.4km | 20 mčs | " |
| 4 | 2 | 2020-01-04 13:S3:03 | 23.4 | 40 | " |
| S  6 | Z  3 | 2020-01-08 21:10:57  nul | 10  nul | 1S  nul | Reslaurant CaxeBtc›n |
| 7 | 2 | 2020-0 -08 2›.30:45 | 25zm | 2S | ixil |
| 8 | 2 | 2020-o1-1o fXf:1S:o2 | 23.4 km | 15 rrl¥x/te | rxil |
| 9 | 2 | nul | nul | nul | Cust‹xrw Caixzgalxxi |
| t0 | 1 | 20Zo-o1-t1 18:s0:20 | t0bzi | l0rraMes | rng |

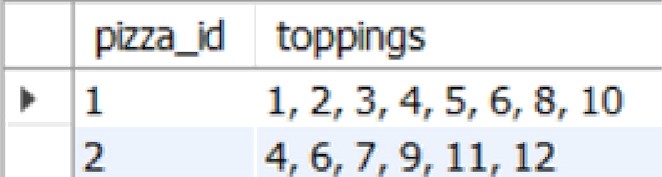
pizzo\_nomes:

At the moment - Pizza Runner only kos 2 pizzos available the Meot Lovers or Vegetorion!



pizza recipes:

Each pizza id has a standard set of toppings thot ore used as port of the pizza recipe.



—

k 1

2

topfXnqs

1, 2, 3, 4, 5, 6, 8, 10

4, 6, 7, 9, 11, 12

pizzo\_toppings:

This table contains all of the topping\_nome values with their corresponc!ing topping\_id value.

|  |  |  |
| --- | --- | --- |
|  | tqppng\_d | topp$›g\_name |
| › | 1 | Bacon |
|  | 2 | x |
|  | 4 | Cheese |
|  | 5 | Clken |
|  | 7 | Orñxis |
|  | 8 |  |
|  | 9 | Peppers |
|  | 10 | S arrg |
|  | 11 | Tomatc›es |
|  | 12 | Tomato Sauce |

Data Cleaning was a bit tricky for this pro|ect task. I have hanc!lec! the missing and null values of tables (i.e., runner\_orders and customer\_orders) in orc!er to clean the c!ata.

customer\_orders:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ordered | cussxner | ¥i | pizza d | excusbm | extras |  |
| r 1 | 101 |  | 1 |  |  | 2920—01-O1 18:05:02 |
| 2 | 1 |  | 1 |  |  | 2020-01-01 19:00:52 |
| 3 | 102 |  | 1 |  |  | 2020-01-02 23:51:23 |
| 3 | 102 |  | 2 |  |  | 2020-01-02 23:51:23 |
| 4 | 1OZ |  | 1 | 4 |  | 2020-01-iH 13:23:46 |
| 4 | 103 |  | 1 | 4 |  | ?0?0-01-04 13.23:46 |
| 4 | 10Z |  | 2 | 4 |  | 2020-01-04 13:23:46 |
| 5 | 104 |  | 1 | nul |  | 2020-01•08 21:00:29 |
| 6 | 101 |  | 2 | nul | nul | 202P-01-08 21:03:13 |
| 7 | 105 |  | 2 | nul |  | 202001-08 21:20:29 |
| s | 102 |  | 1 | nul | nul | 2020-01-09 23:Sd:33 |
| 9 | 103 |  | 1 | 4 | 1, 5 | 2020-01-10 11:22:59 |
| 10 | 104 |  | 1 | nul | nul | 2020-01-11 18:34:49 |
| 10 | 104 |  | 1 | 2, 6 | 1, 4 | 2020-01-11 18:H:49 |

upoate customer orders am€ exclusions : null rheme exclusions : ’ ';

update custornen\_or’ders set exe1us 1ons a null

where etclusioms = "null”;

update customer\_orders set ext ras - nulL whoro extras : '';

updmtc customer\_orders set extras : null whcw• extras = ”null“g

order\_d Dstorr›er\_d $¥zza\_d exd ans extras

> 1 101 1 " "

order\_tzne

202001 01 18:05:02

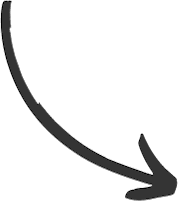


|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2 | I01 | 1 |  |  | 2020-01-01 19:00:52 |
| 3 | 102 | 1 |  |  | 2020-0t-02 23:51:23 |
| 3 | io2 | 2 |  |  | 2020-01-02 Z3:51:23 |
| 4 | 103 | 1 |  |  | 202&-01-04 13:23:46 |
| 4 | 1D3 | 1 |  |  | 2020-01-0d 13:23:46 |
| 4 | 103 | 2 |  |  | 2020-£i1-04 13:23:46 |
| 5 | 1.04 | 1 |  |  | 202001-08 21:00:29 |
| 6 | 101 | Z |  |  | Z0Z0-O I-08 21:03:13 |
| 7 | tO5 | 2 | " |  | 2020-0\-08 22:20:29 |
| 8 | 1D2 | 1 |  |  | 202fh'01-09 23:54:33 |
| 9 | 103 | i | 4 |  | 2020-£i1-10 11:22:59 |
| 10 | 104 | 1 |  |  | 202D-01-11 18:34:49 |
| 10 | l0'l | 1 | 2. fi | 1. 9 | 2020-£i1-LI 18:3't:49 |





|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| before  runner\_orders: | | | | | | |
|  | order\_cI | runner\_d | piCkup\_ome | datance |  |  |
| » | I | I | 2020-01-0118:15:34 | 20km | 32minutes |  |
|  | 2 | 1 | 2020-01-0119:l0:S4 | 20km | 27 Minutes |  |
|  | 3 | 1 | 202O-0I-0300:lZ:37 | 13.4km | 20mins |  |
|  | 4 | 2 | 2020—O1—04 13:53:o3 | 23.4 | 40 |  |
|  | S | 3 | z0z0—0 1 08 21: 10:57 | 10 | 15 |  |
|  | 6 | 3 | nul | nut | n‹uI | Restaurant Cancetaton |
|  | 7 | 2 | 2020-0 1-08 21:30:45 | 25km | 25a | nut |
|  | 8 | 2 | 2020-01-10 00.15:02 | 23.4 km | 15 min‹ute | nul |
|  | 9 | 2 | nul | nul | unul | Customer Cancelaton |
|  | 10 | 1 | 2020-01-11 18:50:2o | 10krn | t0minutes | nul |

upds£ e r•unng'o ander s

\*et pickup\_tSm'e z null wherv exclusions: "n‹ 1.. ' ;

updo£e nunner\_ander r

set distance z nuJi

wherc distance : " l l”;

upds\*e punner\_orders set duration z nuJl

where duration : "ni‹ll”,

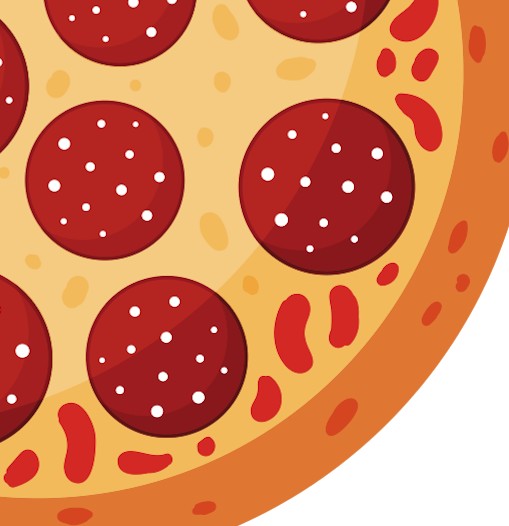
update runner„orders set cancellation : null

where cancellation : "n I ”;

update runne.r\_orders set cancellation : null where cancellation : ";



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| order\_cI | runner\_d | pckup\_t?ne | distance | duration | cancetatbn |
| * 1 | 1 | 2020-01-01 18:15:34 | 20 km | 32 mn |  |
| 2 | 1 | 2020-01-01 19:10:54 | 20 km | 27 mn | ” |
| 3 | 1 | 2020-01-03 00:12:37 | 13.4 km | 20 mn |  |
| 4 | 2 | 2020-0l-04 13:53:03 | 23,4 km | 40 mn | ” |
| S | 3 | 2020 01-08 21:1O:S7 | 10 km | IS mn | ” |
| 6 | 3 | ” | ” | ” | Restaurant Cancetation |
| 7 | 2 | 2020 01 08 21:30:4S | 2S km | 2S mn | ” |
| 8 | 2 | 2020-01-10 00:15:02 | 23.4 km | l5 mn |  |
| 9 | 2 |  | ” | ” | Customer Cance0at tn |
|  | 1 | zozo-of-11 IB:so:zo | to kr» | to mn |  |





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

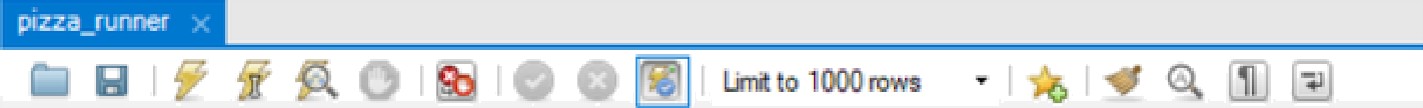
* Pizza Metrics
* Runner and Customer Experience
* Ingredient Optimisation
* Pricing and Ratings

Each of the following case stucly questions can be answered using a single SQL statement.



1. How mony pizzas were ordered?

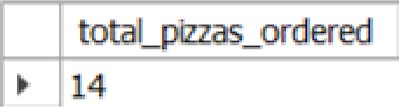


* 1. • se1ect count(•) as tata1\_p1zzas\_ordered
  2. fw<xe custoner\_or•dersy

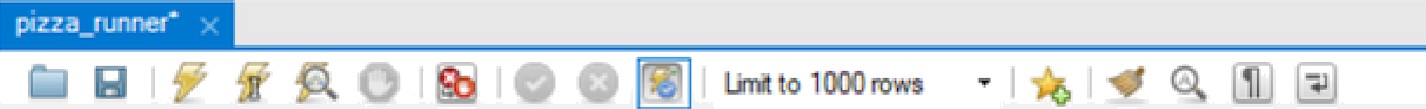
3

4

‹

aaane+u e•a a w c«s

1. How mony unique customer orders were mode?



* 1. • salect count (d€at£nct custo‹aer‘\_1d) aa tota1\_custoaer s
  2. from custofeer’\_ordersy

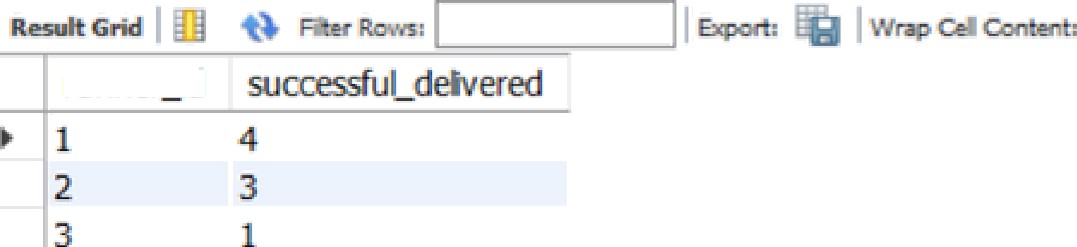
4



1. How mony successful orders were delivered by each runner?



* 1. • select runner\_1d count(^} as sueces sfu1\_de1lvered
  2. from r'unner\_order's
  3. where cance1 fat aon is nulE



runner d

1. How mony of each type of pizza was delivered?
   1. • select p1zza na4›e, coun t( \*) as pt zza del 1vered
   2. froe custozser\_orders c
   3. 1aft •joln p1z z a\_nazses p usxnjj(p1z za\_1d)

d wh•r• order\_id zn (

**salact** order\_1d

fr•m pi z z a\_runner - runner\_onders wh•r• cance11at:ton 1s null)

8 group by 2 j



### How mony Vegetarian and Meotlovers were ordered by each customer?

e r r D u m ‹•›• - ” x < a tm e

1. • sa1rct customer\_id pi z za\_name count (\*) as pi z za\_o dered
2. f'rom custoeer\_orders c
3. joln pt z za\_naues p us1ng(pt zza„1d}
4. group by 1 2 w1th rollup

2 hav1ng p1z z a\_naae ts not nuSI j

|  |  |  |
| --- | --- | --- |
| * tOt | ueatbvers | 2 |
| 101 | vegetarian | l |
| 102 | l4eatbvers | 2 |
| 10Z | ve9etarBn | l |
| 103 | vegetarian  vegetarian | i  i |

1. What was the maximum number of pizzas delivered in a single order?

I • — wit h del1vered as ‹

1. ceJect order id count( \*) as pizza de1 ivered
2. from cust omer\_orders
3. whare order’\_1d 1n (



6

7

1. group by 1}

sclcct order\_Sd

from pizza\_runner.runner orders whmrm cancellation iz null)

1. select max(p1zza\_de1ivered) as max\_pi zza„de1free IB froe del1vered j



* 3

1. For each customer, how mony delivered pizzas had at least 1 change and how mony had no changes?



* 1. • ca 2ack custoaer\_id,
  2. sum(<ae• uh•n exe1us1ons Sa nu21 and extras 1s null then 1 e1c+ 6 and) a s no\_changes
  3. sum(caae when {exclusions I.a nu'I1 and extras Ie not null) or

(exe lus i ones £« no' null and extras 1> null) or

1. {exe3usSons L• n•t null •nd extras 1• n-I nu12J h•n 1 •1•• 8 and) •• at1ee st\_one\_change
2. »\*cr• order\_id \*u (



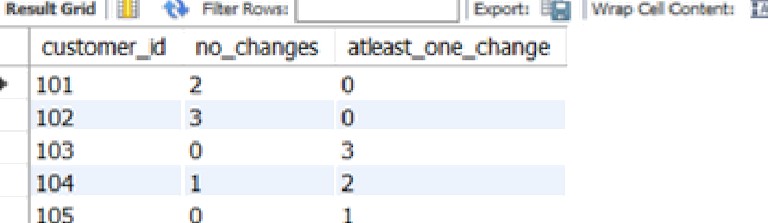
9

10

o•Joct order id

\* ••' pit z+\_ruaaer . naae.»\_oWe»s

wh•r• concell^tiOu £m null)



1. How mony pizzas were delivered thot had both exclusions and extras?



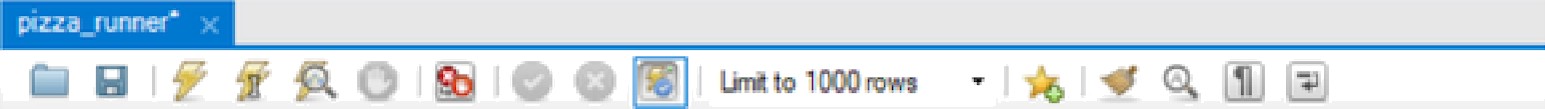
* 1. • se2act suo{casa when exclusions Le nod null and extres ta nod nu13 then 1 end) aa no of lzzes
  2. fraei custc›a›er\_orders
  3. - uhar• om;Ier\_1d In (

celact order\_id

5 fr oo pt zzap.runner. runner\_orders wh are caneaLI at hon be null)

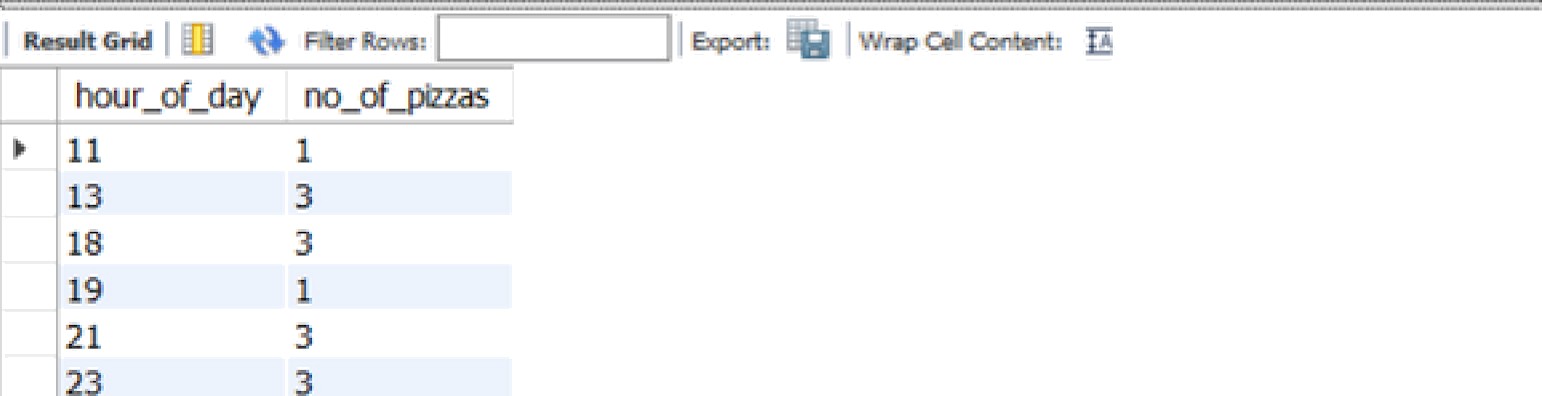


1. What was the total volume of pizzas ordered for each hour of the day?

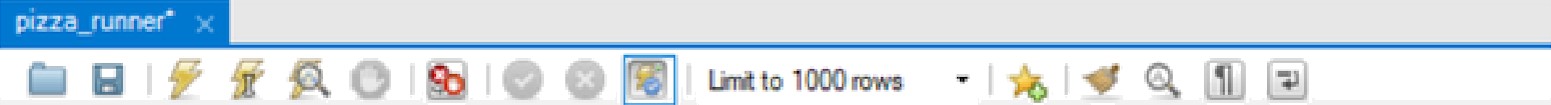


3 • **selact hou¥•(order\_ti«›e)** aa **hour\_of\_dey** count ( \*) •a np\_of\_pi zzgs

1. fz•om cust olae r' order•s
2. gr'oup b'y 1 or'dar by 1j



JO. What was the volume of orders for each day of the week?



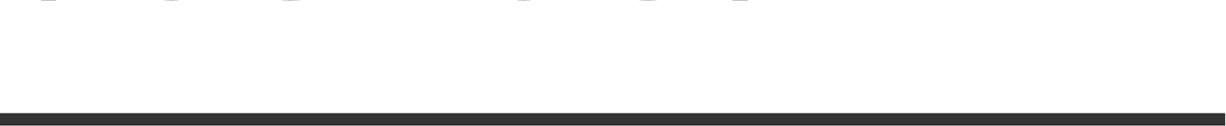
1. • aa1act dayofweek(order\_t1me) •s day\_nun,
2. ayname(onder\_t1ne) as day\_of„ueek, count(\*) aa no\_of\_p1zzas

ñ front custoner\_orders





|  |  |  |
| --- | --- | --- |
| day\_num | day\_of\_week | no\_0f\_pIzzas |
| ›s 4 | WedresdaV | 5 |
| ;! 5 | Thursday | 3 |
| ’! 7 | SzgzJrday | 5 |

INSIGHTS GATHERED

* A total of 14 pizzas were ordered.
* A total of 5 unique customers ordered pizzas.
* The number of successful orders was eight with which Avnner I has the fiigfies/ number of successful orders delivered.
* Meatlovers lype of pizza was sold nine unifs whereas Vegetarian was

sold three units.

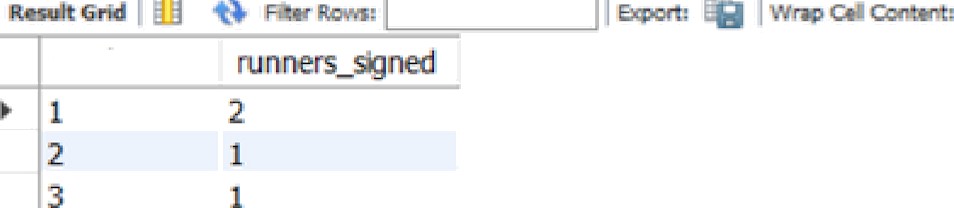
* In a single order, the *maximum* number of pizzas delivered was three.
* One single pizza was delivered that kad botk exclusions and extras.
* Wednesday has delivered the highest number of pizzas.



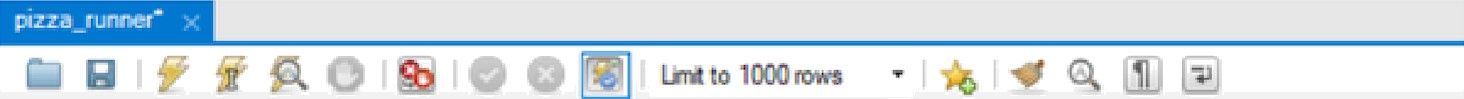
1. How mony runners signed up for each 1 week period? (i.e. week starts

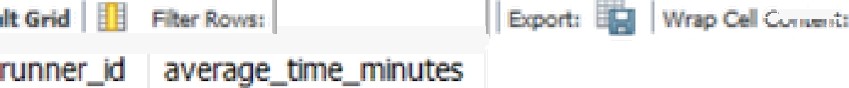
## 202J -OJ -OJ)

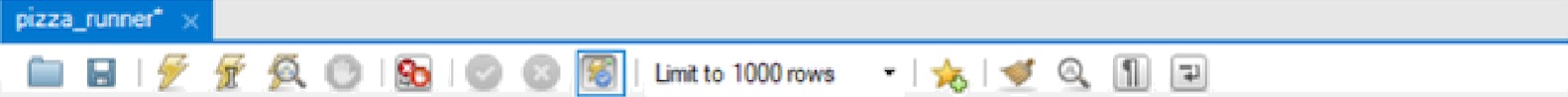
* 1. 8 zalact weeko fyea n(r•eg1stret?on\_dete + ñnt•rva1 1 u••k) •a ueek\_nun count {8 ) ao runn.ers\_szgned
  2. from runners

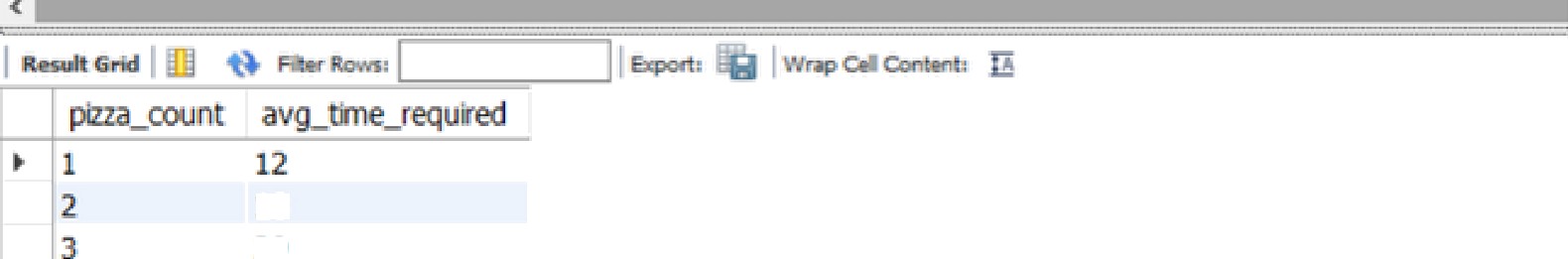


weeIt\_num

1. What was the average time in minutes it took for each runner to arrive at the Pizza Runner HQ to pickup the order?
   1. \* ”-'” ulth avgt1•e aa (
   2. zeI•ct d1ztlnct r •order 1d r • runnei \_id
   3. **1-»t•mpd1ff(nSnut•** or^der\_txne p1ckup\_t\*zae) •s m;tnwtes
   4. fz•om runner\_orders r
   5. 1nn•r yoln custoner\_orders c using(or'der\_\*d)
   6. vshere cance1tation. 1s null)
   7. select runner\_1d , round(avg(n1nutes}) as average\_t ifoe\_n1nutes
   8. f'ron avgt1•e



1. Is there any relationship between the number of pizzas and how long the order takes to prepare?
   1. • select p1zz a\_count round(avg(t1fae\_requ1red}) a.s avg\_t1me„requ1red
   2. ' re«t ( select d't st1nct r .order\_1d,
   3. t1atestampd1ff(m:Lnute, order\_t1›me, pt ckup\_t lose) as I 1rse\_requ1red,
   4. count ( • ) over(pai•t1t1on by order\_id) as pt zza\_count
   5. froa runner\_orders r
   6. Snner join custoaer\_ord!ers c us1ng(order\_1d )
   7. where cance11at ton xs null) as re1at ton
   8. group by J y



1. What was the average distance travelled for each customer?



* 1. • select cust oaer\_źd sound( avg{dž st am e) , 1) as avg\_dż stance
  2. runner\_orders r
  3. jożn ć ust omer'\_orders c us1njj(or'der\_1d)
  4. where d1st anCe żs not null
  5. group by 1 order by 1j

|  |  |
| --- | --- |
| so 0l ‘t¥  aJstomer\_Ò | :  avg\_dòtanœ |
| D 101 | 20 |
| 102 | 16.7 |
| 103 | 23.4 |
| 104 | 10 |
| 105 | 25 |

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



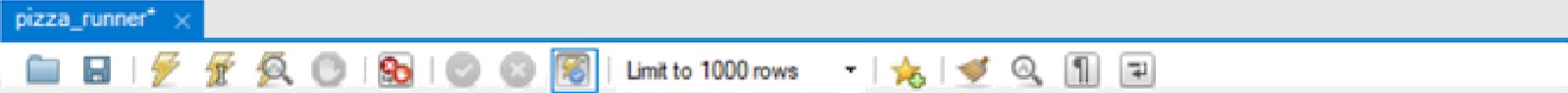
* 1. • select max(duration) as longest\_delivery\_times,
  2. rin(duration) as shortest delivery\_time, (mcx(duration) (duration)) as difference

4 from runner\_orders;

detvery\_trnes shortest\_dØvery'\_tme deference

> 40 mh 10 rnh 30

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



* 1. • — u1th average as

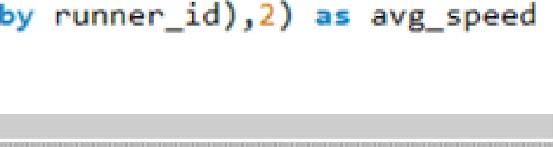
a•1•ct order\_śd, runner\_id›

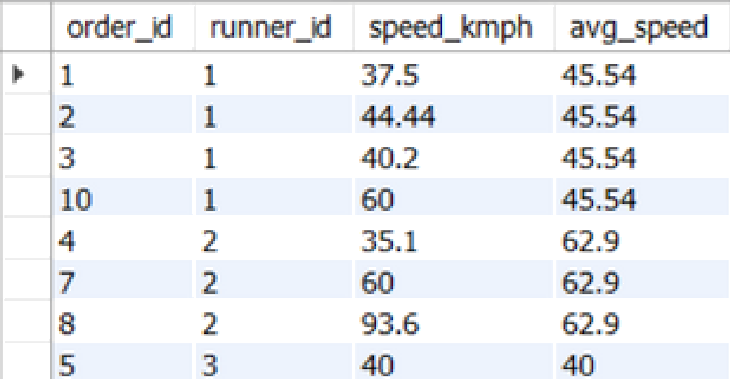
round( (distence \* Ö0 ) /durøtion, 2 ) •ø speed\_kmph

from r‘unne r\_o nde s

whorc cancellation io mull

b 

”ø nuund(avg( speed\_kmph) average j

œ+i ra+r . : wt«o cø ac- G

|  |  |  |  |
| --- | --- | --- | --- |
| *oÆef\_”b* | rtmrier\_Õ | speed\_kmph | avg\_speed |
| \*' 1 | 1 | 37.5 | 45.54 |
| 2 | 1 | 4s.44 | 4S.S4 |
| Z | 1 | 40.2 | 45.54 |
| L0 | 1 | 60 | 45.S4 |
| 4 | 2 | 35.1 | 62.9 |
|  |  | 93.6 | 62.9 |

1. What is the successful delivery percentage for each runner?



with percentage zm C n•l•ct xnner\_id,

sum(c^cu ‹Whu« caacelJaCion ia nwll th•n 1 cln• 0 •nd) ^s succ#ssCul delivery, count(°) zs total\_ord•r

" from nunner\_ordero

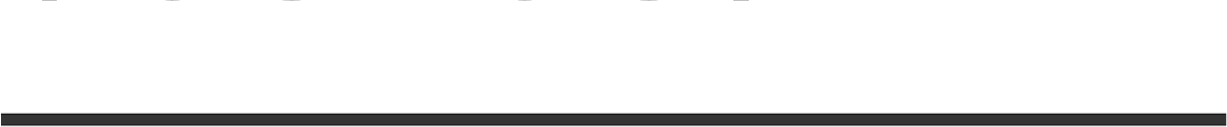
6 6"°°P\*Y \*l

C concat (round( (succassfu1\_da11vary\*100/tota1\_order) ,2 } " g" ) as soccas sfu1\_daJ:ivary\_prcnt

9 A rots percentage j



2 75XXl %

INSIGHTS GATHERED

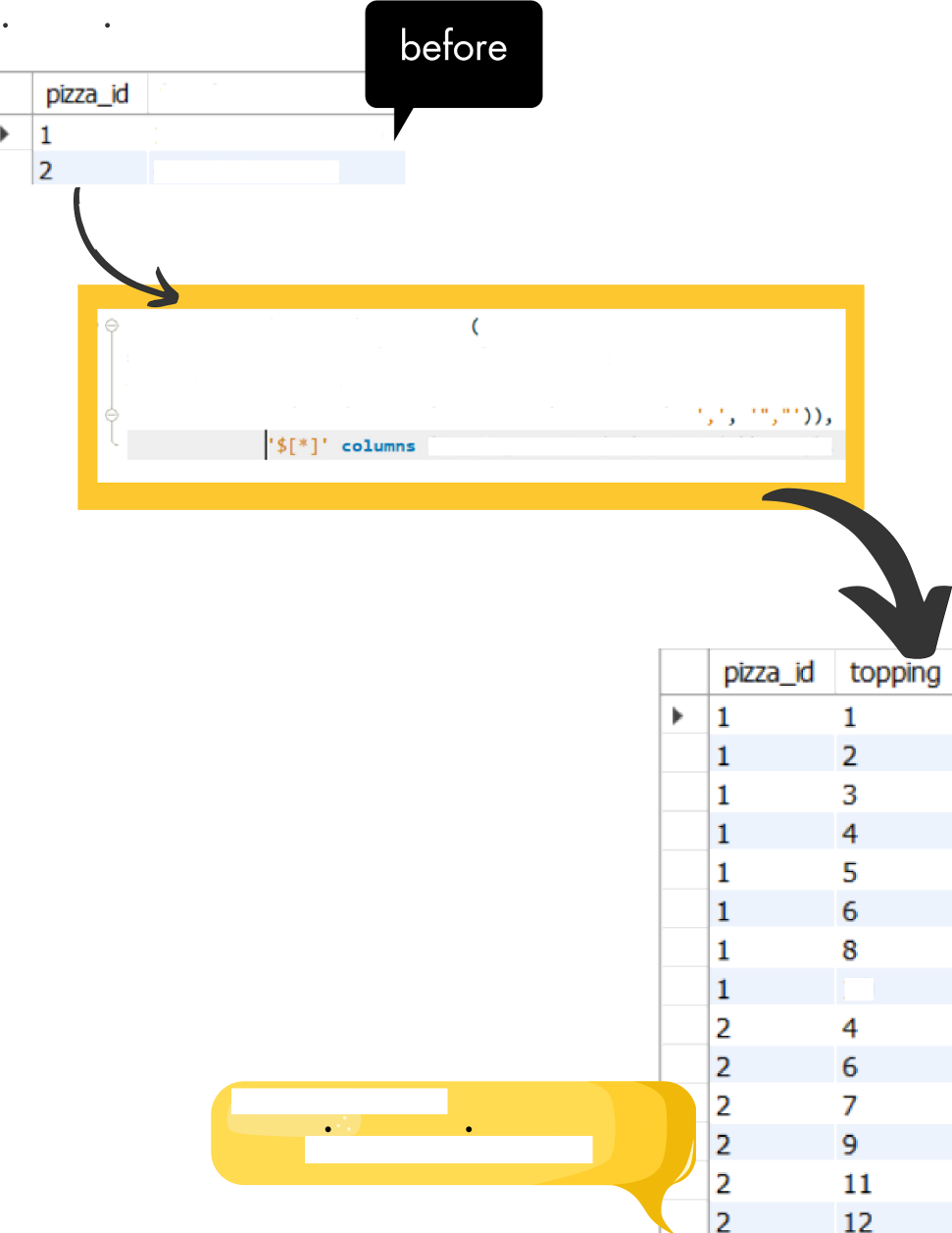
* In the first week, there is the *highest* number of runners were signed.
* Pizza Runner 2 takes a *longer* time whereas Pizza Runner 3 takes the

*shortest* time to arrive at Pizza HQ to pick up the order.

* The relationship between the number of pizzas and the time required to prepare is positively correlated.
* The difference between the *longest* and *shortest* delivery times for all orders was 30 minutes.
* Pizza Runner 1 has the *highest* successful delivery percentage.



*So, to* gather information about the ingredients we need to transform the tables (i.e., pizzo\_recipes and customer\_orders) and make them as a virtual table (VIEW based on the result-set of an SQL statement.



pizzo\_recipes:

1, 2,3, 4, 5, 6, 8, 10

4, 6, 7, 9, 11, 12

z••zt• vi•w pizza\_recipe\_new z•

re\*ect r.pizza\_id, trim(j.topping) cs topping

from pizza\_recipes r

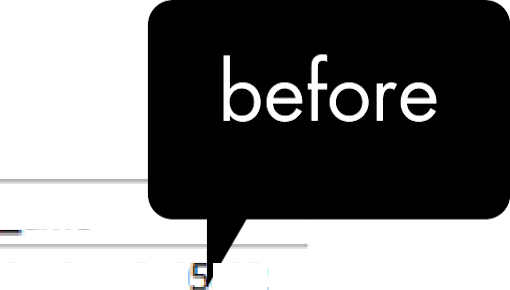
join j son\_tab1e(I r1m( relpface( j son\_array( r . toppings},

(topp1ng varchar(se) paCh ' $ ' )) as §} j

10

## e ew as

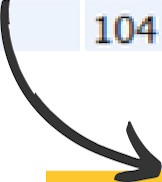
pizzo\_receipe\_new

customer\_orders:

‘

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| order\_d | customer\_d | pizza\_i1 | exc:hsöns | extras | order üme |
| > 1 | 10J | 1 | ” | ” | 2020-OR-01 18:0 !02 |
| 2 | 10J | 1 |  |  | **2020-01-0t** 19:00:52 |
| 3 | 102 | I |  |  | 2020-01-02 23:51:23 |
| 3 | 102 | 2 |  |  | 2020-01-02 23:51:23 |
| 4 | 103 | 1 | 4 |  | 2020-01-04 13:23:46 |
| 4 | 103 | 1 | 4 |  | 2020-01-04 13:23:46 |
| 4 | 103 | 2 | 4 |  | 2020-01-04 13:23:46 |
| S | i04 | i |  | 1 | 2020-01-08 21:00:29 |
| 6 | 101 | 2 |  |  | 2020-01-08 21:03: f3 |
| 7 | 105 | 2 |  | 1 | 2020-01-08 21:20:29 |
| 8  9 | 102  103 | 1  1 | 4 | i, s | 2020-01—09 23:54:33  2020—OU—10 11:22:59 |
| lo | iß4 | 1 |  |  | 2020-01-11 18:34:49 |

‘

10 1 2, 6

1,4

2020-OU-11 18:34:49

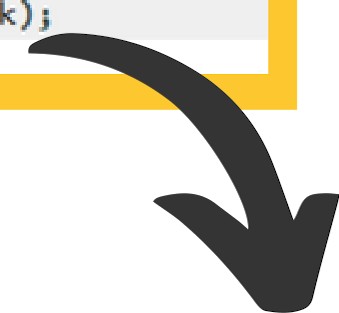
create view customer order new a s (

**select** c.onder Id, c . cu st omer\_id , c . pi z z a id, I i ’ ( j . excl us ion s) as exc1us i an

t' i'’(k.extras) a& extras, order\_time Crom customer\_orders c

join j\*on\_tzble(t’i (rvplzce(json\_array(c.exclus\*ons), ,', "',” )),

I ’ column\* (exclusions vnrchor(-t) path 'S’)) n\* j join j\*on\_tzb1e('' in(replocv(json\_array(c.extras), ,’, ')),

S ’]' columns (extras varchar(50) path 'S')) ao

order\_Ö customer\_cI pcza\_<I exclusion extras

r 1 101 1 ” "

order\_txne

Z020-01-01 18:05:02

# cusDmer\_order\_new

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 2 | 1 | 1 | ” | ” | 2020-0v0l19:00:S2 |
| 3 | 102 | 1 | ” | ” | ]020I1-0223:51:]3 |
| 3 | 102 | 2 |  |  | 2020-01.0223:S1:23 |
| 4 | 103 | 1 | 4 |  | Z020I1-O4 l3:23:46 |
| 4 | 103 | 1 | 4 |  | 2020-01-04 13:23:46 |
| 4 | 103 | 2 | 4 |  | 2020-01-04 13:23:46 |
| 5 | 104 | 1 | ” | 1 | 2020-01-08 21:00:29 |
| 6 | 101 | 2 | ” | ” | 2020—0I—08 21:03:13 |
| 7 | 105 | 2 |  | 1 | 2020-01-0821:20:29 |
| 8 | 102 | 1 |  |  | 2020-0I -09 23:54:33 |
| 9 | **103** | 1 | 4 | 1 | 2020D1-1011:22:59 |
| 9 | 103 | 1 | 4 | 5 | 2D2O-01 -10 11:22:59 |
| fo | 104 | i |  |  | 2020-01-11 18:34:49 |
| 10 | 104 | 1 | 2 | 1 | 2020-01-11 18:34.49 |
| 10 | 104 | 1 | 2 | 4 | 2020—01—11 18:34:49 |
| 10 | 104 | 1 | 6 | 1 | 2D2D-01-11 18:34:49 |
| 10 | 104 | 1 | 6 | 4 | 2020—01-11 18:34:49 |
| create view as |  |  |  |  |  |  |

1. What ore the standard ingredients for each pizza?



l • selott p2.pizza\_ngme,

- ( p 3. to pp i ng\_n ame s epo • o I er ' ) o: standard\_ingredisnts

from gizza\_recipe\_new pl

inncr join pizza\_names p2 u-mq(pizza\_id)

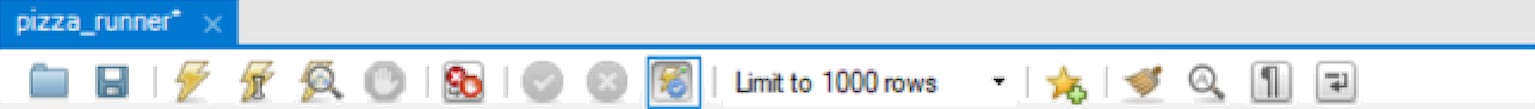
6 inncr join pizza\_toppings p3 on p3.toppìng\_id = pS.toppìng



pRza name standard +mg edenn

r Meaoovers hon, BBgi Sauce, Beef, Cheese, Chcken., I \u5hroonJs, Pepperoni, SaBmi VeqeCanan Cheese, Mushrooms, Onions, Peooers, Tomatoes, Tomato sauce

### What was the most commonly added extra?



l • select topping\_name zs most\_commonly\_added\_extra from pizza toppings

\vhe re t opp i ng\_z d - (

select extras

select extras, t(\*) zs times\_used from customer\_order\_new

 where extras is not null

group Iby I order by desc limit :)

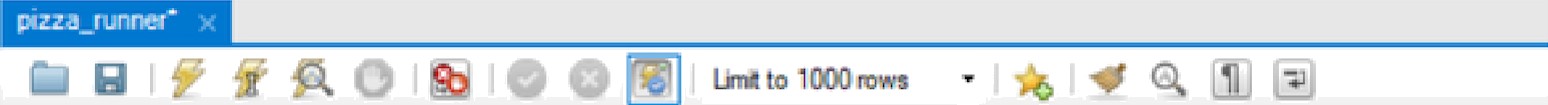
a s t ) j

 np a q :va ce ccter. r

most commonly added extra

› Bacon

### What was the most common exclusion?



1 - :olecC topping\_name s most\_common\_exc1usion



'.'.'h e we I opp1ng i *d --* (

 :cleut exclusion

so I o ct exc I us ion , ( \* } a» I i me s\_us ed

from Customer\_order\_new

:hcrc exclusion i, not null

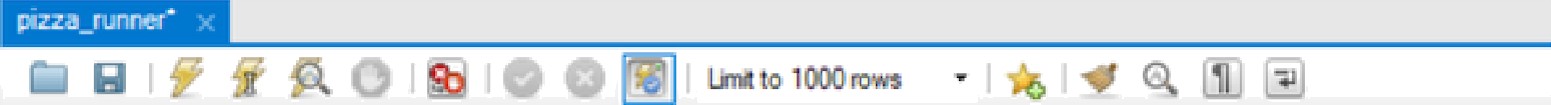
group b; I o dcr by disc limit l )

Result G nd 

most\_common\_exctision

› Cheese

1. Generate on order item for each record in the customers orders table in the format of one of the following:
   * Meat Lovers
   * Meat Lovers - Exclude Beef
   * Meat Lovers - Extra Bacon
   * Meat Lovers - Exclude Cheese, Bacon - Extra Mushroom, Peppers.



* + 1. ^ - w1th generate cust order as (
    2. a a fact d1st met con. order\_id, con . custooer\_id, con . pi zz a\_1d ,

\* can c at ”Ex c I ude ”, pt. t opping\_nene) as exe1us ton

con. extr'as , con.or’der\_tzme

from cuStomer\_or'der\_new con

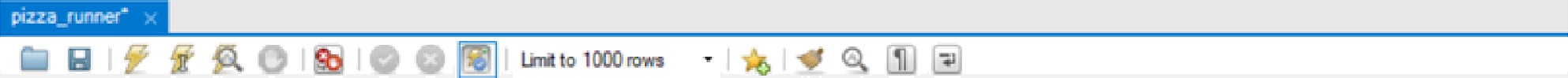
6 left join pizza\_toppings pt

on con. exc1us1on - pt . I opp1n x d

1. se2ect d1st in ct gc . order\_1d, gc - cust omer\_1d, gc. pt z za\_1d,
2. gc . exe1 usion as exe1usion
3. c can c a t ( " E xt r a " J pt - t op ping\_n sr›e) ss ext ras gc . rder•\_t inc
4. from gener'ate\_cust\_or’der gc
5. left join p1zza\_topp1ngs pt
6. **on gc.extras** pt.topp1ng\_1d;)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| order\_xI | custorner\_id | péza\_d | exctzsbn | extras | order\_szie |
| b 1 | 101 | 1 |  |  | 2020-01-01 l8:05:02 |
|  |  |  |  |  | 9030<1Ill9:OO:S2 |
| 3 | lD2 | 1 |  |  | 20Z0-01OZZ3:'51:Z3 |
| 3 | 102 | 2 |  |  | 2020I1T223:SLl3 |
| 4 | 1D3 | 1 | CxtfJde Cl-ieme |  | 2020-01-00 13:Z3:Q6 |
| 4 | 103 | 2 | CxcfJde Cheme |  | 2020&1-04 13:ZZ:46 |
| 5 | 104 | 1 |  |  | 20Z0-01A8Zl:D0:Z9' |
| 6 | 101 | 2 |  |  |  |
| 7  9 | 105  103 | 2  1 | Cxckxie Cheese | Cxtra 8acon | **2D20—01—08** 21:TO:29  2020-01-09 23:54:33  2020-01-10 11:22:5g |
| 10 | lW | 1 |  |  | 2020W1-10 11:22:59•  2020T1-1118:34:49 |
| 1o | 104 | 1 | ExcLdeBBQSauce | ExbaBacon | 2020-01-11 1B:34:49' |
| 1o | 104 | 1 | McLxteBBpSauce | ExaaCheese | 2020-01-11 1B:34:49 |
| 10 | 104 | 1 | McLdeNushr ms | ExbaBa¢on | 20?0-01-11 1B:34:49' |
| 10 | 1D4 | 1 | ExcLdeNushr ms | ExbaCheese | 2020—O1-11 1B:34:49 |

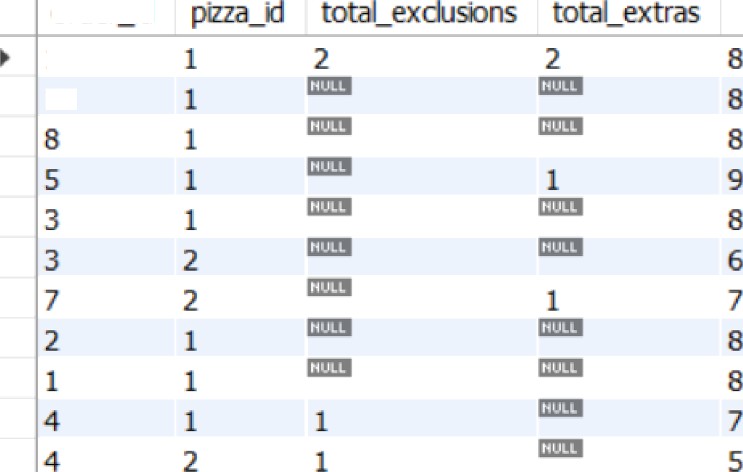
### What is the total quantity of each ingredient used in all delivered pizzas sorted by most frequent first?



1 • '- w1th updat\*on •» (

|  |  |  |
| --- | --- | --- |
| 2 |  | zmlmc€ diztinc€ c.OwdQo\_id, duration, c.piaza\_Sd, |
| 3 |  | length(o.Qx&lusSonsj • length(rmplncm(c.exclusions, ' ,', '')) + 1 ns total\_exclusions, |
| 4 |  | length(c.extras) - length(rcplcee(c.extras, ',', ’')) t 1 ms total\_extras, |
| 5 |  | length(p.toppings) - length(neplo\* (p.toppings, ’,', ’ ’)) + l •\* totzl\_topping’s |
| 6 |  | Crmn runner\_orders r |
| 7 |  | join custcxner\_orders c usSng(order\_1d) |
| 8 |  | join pz z za\_rec Apes p on p. p1xza\_1d z c . pt z za\_1d |
| 9 |  | whara r. cance11et.ton 1s n.ull |
| ie |  | «w•r by duration aac} |
| 11 |  | sa1•ct a.rder•\_1d p1zza\_1d tota1\_exc1us 1ons, tota1\_extras, |
| 12 | ' | casa than I:ota1\_exc1usions = tota1\_extr‘as then tote1\_topp1ngs |

1. when I ota 1\_exc1usions 1s null and tot a1\_ext ras 1s null then I ota1\_topp1ngs
2. whan toCa1\_exc1usions 1s null and tot a1\_ext ras ›o i then (tota1\_topp1ngs + tot a1\_extras)
3. alse {tota1\_topp1ngs - tota1\_exc 1usions )
4. and as topp1ngs\_requ1red
5. froa updatzonj



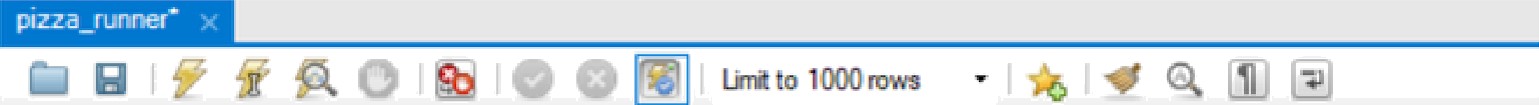
order ci

iO

I NSIGHTS GATHERED

* The most commonly ac!c!ec! extra was "Bacon”.
* The most common exclusion was "Cheese”.
* The maximum number of toppings ac!ded was nine.

1. If a Meat Lovers pizza costs $ \ 2 and Vegetarian costs $10 and there were no charges for changes - how much money has Pizza Runner mode so for if there ore no delivery fees?



l ° with total as (

2 with psi z za\_detai1s a x ( select •

trom runner\_orders

where cance11at 1on 1s null}

6 salect d1st1ncI ” 1f(p1zza\_1d = 1, 12, 10} as pi z za\_cost

from• pizza\_details pd

8 joln cu stomer\_orders co using(order\_1d))

9 s•l•ct runner\_id, sum(pizza\_cost) os total\_earning

16 from total

11 group fry 1

nJnner\_d totd\_eamrig r 1 70

### What if there was on additional $T charge for any pizza extras? o Add cheese is $1 extra

e r x a u a ‹• ••- - a < o a -

1 •- with addIt 1ona1\_det at 1s as (u1th p1z za\_det ai I s as (

salact

from r‘unner\_orders

vhomo cancellation ir null)

select d1s t1nct \* 1T(p1zza\_1d = 1, 12, IO) •s pt z za\_cost , c•se uhen co. extras 1s null then 0

else { 1er gt h(co. ext ras) 1- ngI I ( replace(co. ext ras, ’ , ' end as add1t i ona1\_cost

Unani pt z za\_det ai 1s pd

yoln cus tomer\_orders co us1ng(order\_id})

oslsct runner\_id, sue(pizza\_cost+sdditienel\_cost) zc totel\_cost

12 from additional\_detaSls

' ' ) ) + 1}

|  |  |
| --- | --- |
| ”1 | 72 |
| 2 | 45 |
| 3 | 13 |

1. The Pizza Runner team now wants to odd on additional rating system that allows customers to rote their runner, how would you design a supplementary table for this new dataset - generate a schema for this new table and you con insert your own data for ratings for each successful customer order between 1 to 5.

creata vlaw runner\_orders\_new as (

elecC °, if(cancellation is null, floor(I + rand() ° 5), null) cs catiugs

from runner\_order s j

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| order\_d | runne id | @kup time | dBtance | duratbn | cancetaton | ratñgs |
| 1 | 1 | 2020-D1 DI 18:15:34 | 20 km | 32 mn |  | 3 |
| 2 | 1 | 2020-01-0119:t0:54 | 20km | 27mn |  | 4 |
| 3 | 1 | 2020-01-0300:12:37 | 13.4km | 20mn |  | 5 |
| 4 | 2 | 2020-01-0413:53:03 | 23.4krn | 40mn |  | 3 |
| S | 3 | 2020-01-08 21:10:57 | 10 km | 15 md |  | 3 |
| 6 | 3 |  |  |  | Reslaurant Care:e3atbn |  |
| 7 | 2 | 2020-01-08 21:30:45 | 25 km | 25 md | " | 1 |
| 8 | 2 | 2020-01-10 00:15:02 | 23.4 len | 15 mtt |  | S |
| 9 | 2 |  |  |  | Customer Caix:elation |  |
| lo | 1 | 20Z0-01-11 18:50:20 | 10 diri | l0 MPi |  | 4 |

1. If a Meat Lovers pizza was $\ 2 and a Vegetarian $10 fixed prices with no cost for extras and each runner is paid SO.30 per kilometer traveled - how muck money does Pizza Runner have left over after these deliveries?



* 1. °— with leftover as (
  2. with total as(

solmct distinct c.order\_id, r.runner\_id, c.pizza\_id, r.distance,

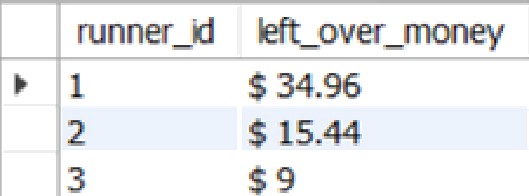
1. i€(c.pizza\_id : l, 12, 18) nm pizza\_jprice, (r.distance"0.30) ns runner\_paid

from c ustomer\_orders c

josn r'unner\_or'ders r' using(order\_id) where cane e1 fat ion 1s nu11$

8 seEect runner\_id, sum(p1zza\_pr:tce) as I ota1\_piz za\_price, sum(r'unner\_patd) as tot aL\_runner aid

16 from totaL

1. group by I )
2. select runner'\_ d
3. con cat "g " round {total izza r‘1ce - I ota1\_r•unner aid 2} ) as 1eff\_aver\_maney
4. fram Leftover;

|  |  |
| --- | --- |
| runner\_id | ft\_pver\_money |
| \* t | S 34.96 |
| 2 | S t5.44 |

1. Using your newly generated loble - con you join all of the information

together to form a table that has the following information for successful deliveries?

o customer id

0 order id"

0 runne id

# 0 rdnr time

0

o pickup\_time

o Time between order and pickup

0 Delivery duration

0 Average speed

0 Total number of pizzas



z•on•pmcktqt\_tzee,

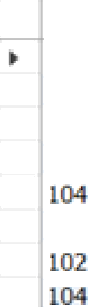
I oncac(44••at•a6dA+ faza•t•e con•o• ier\_t:Iae •p1ckx +\_tSee}," •Sn") •• t1ee\_d1fF\_ordee\_md„ptck ,

1. non.z¥zrataan •s de11vcry\_durat1on round(r'on.d1stanca•6e/non.durat\*om, 2) •a spaad\_kaph,
2. count{\*) wvar{par•t£t5on by r•on.or•dar\_ad) •a tota1\_no\_of\_p1zzas
3. ft custaaar\_or'da¥'\_r+am cou
4. rAgI•t 3•4•• mmner\_ s\_nev ron ••5ngforder\_Sd}
5. uhaz•a c•r+ce12atioai fic null}

IO calact ousC \_1d onder\_1d ruzusar\_sd r•ound{avg{rat1nga)) •a rslz1ngs

1. order\_tTae p1ckzqs\_t1wa txse\_d1ff\_or•dér\_end\_p1ckup
2. deIivsey\_durwtion, \*pe\*d\_W•ph, to4sl\_no\_of\_pizzss

¥3 +-cte



¥ 101

101

102

1

2

3

¥ 1

1

B

1o

2

1

3

L 2

9

+

2

2 I•01-01tB XYZ

•0tt6:15:34 10nm

2tWI•01-0119fBXSZ Z/WMft-OF 1@¥0:H 10¥r€t

a)d0 i1-0223:S€ 3020-0£-03fXt:22:37 21o0i

20d0-III-06 21«¥X29 2020-01-IB 21:10:57

20 0-€11-09 ZZ:54M3 ZO20-01-10 CYL’.15:IIZ

a¥d0-o1-IL LB:B4c4B 2oa6oz-11 zaW•Z0

10 nEi

20 zrM

Isztñs

1s«m

EOIsOs

s

INSIGHTS GATHERED

* Pizza Runner 1 has earned $ 70 among the other two runners if there is

*no extra charge* for the extras.

* Pizza Runner 1 has earned $ 72 among the other two runners if there is

*an extra charge* for the extras.

* Pizza Runner 3 has $ 9 left after all the deliveries.

