

```
1 • create database pizza_runner;
2 • use pizza_runner;
3 • create table runners(
4     runner_id int,
5     registration_date date);
6
7 • insert into runners(runner_id, registration_date)
8     values
9     (1, '2021-01-01'),
10    (2, '2021-01-03'),
11    (3, '2021-01-08'),
12    (4, '2021-01-15');
13
```


14

15 • ⊖ create table customer_orders(
16 order_id int,
17 customer_id int,
18 pizza_id int,
19 exclusions varchar(4),
20 extras varchar(4),
21 order_time timestamp);

- `INSERT INTO customer_orders`
 `(order_id, customer_id, pizza_id, exclusions, extras, order_time)`
`VALUES`
 `('1', '101', '1', '', '', '2020-01-01 18:05:02'),`
 `('2', '101', '1', '', '', '2020-01-01 19:00:52'),`
 `('3', '102', '1', '', '', '2020-01-02 23:51:23'),`
 `('3', '102', '2', '', NULL, '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'),`
 `('5', '104', '1', 'null', '1', '2020-01-08 21:00:29'),`
 `('6', '101', '2', 'null', 'null', '2020-01-08 21:03:13'),`
 `('7', '105', '2', 'null', '1', '2020-01-08 21:20:29'),`
 `('8', '102', '1', 'null', 'null', '2020-01-09 23:54:33'),`
 `('9', '103', '1', '4', '1, 5', '2020-01-10 11:22:59'),`
 `('10', '104', '1', 'null', 'null', '2020-01-11 18:34:49'),`
 `('10', '104', '1', '2, 6', '1, 4', '2020-01-11 18:34:49');`

41 • `select * from customer_orders;`

42

Result Grid   Filter Rows: | Export:  | Wrap Cell Content: 

	order_id	customer_id	pizza_id	exclusions	extras	order_time
▶	1	101	1			2020-01-01 18:05:02
	2	101	1			2020-01-01 19:00:52
	3	102	1			2020-01-02 23:51:23
	3	102	2		NULL	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
	5	104	1	null	1	2020-01-08 21:00:29
	6	101	2	null	null	2020-01-08 21:03:13
	7	105	2	null	1	2020-01-08 21:20:29
	8	102	1	null	null	2020-01-09 23:54:33
	9	103	1	4	1, 5	2020-01-10 11:22:59
	10	104	1	null	null	2020-01-11 18:34:49
	10	104	1	2, 6	1, 4	2020-01-11 18:34:49

customer_orders 1

```
44 ● ⊖ create table runner_orders(  
45     order_id int,  
46     runner_id int,  
47     pickup_time varchar(20),  
48     distance varchar(20),  
49     duration varchar(20),  
50     cancellation varchar(50)  
51 );
```

02

03 •

04

05

06

07

08

09

10

11

12

13

14

15

16

17

18

```

INSERT INTO runner_orders
(order_id, runner_id, pickup_time, distance, duration, cancellation)
VALUES
('1', '1', '2020-01-01 18:15:34', '20km', '32 minutes', null),
('2', '1', '2020-01-01 19:10:54', '20km', '27 minutes', null),
('3', '1', '2020-01-03 00:12:37', '13.4km', '20 mins', NULL),
('4', '2', '2020-01-04 13:53:03', '23.4', '40', NULL),
('5', '3', '2020-01-08 21:10:57', '10', '15', NULL),
('6', '3', 'null', 'null', 'null', 'resturant cancellation'),
('7', '2', '2020-01-08 21:30:45', '25km', '25mins', null),
('8', '2', '2020-01-10 00:15:02', '23.4 km', '15 minute', null),
('9', '2', 'null', 'null', 'null', null),
('10', '1', '2020-01-11 18:50:20', '10km', '10minutes', 'customer_cancellation');

```




```
69 ● ⊖ CREATE TABLE pizza_names (  
70     pizza_id INTEGER,  
71     pizza_name TEXT  
72 );
```

```
73 ● INSERT INTO pizza_names  
74     (pizza_id, pizza_name)  
75     VALUES  
76     (1, 'Meatlovers'),  
77     (2, 'Vegetarian');
```

```
78  
79 ● ⊖ CREATE TABLE pizza_recipes (  
80     pizza_id INTEGER,  
81     toppings TEXT  
82 );
```

```
83 • INSERT INTO pizza_recipes
84     (pizza_id, toppings)
85     VALUES
86     (1, '1, 2, 3, 4, 5, 6, 8, 10'),
87     (2, '4, 6, 7, 9, 11, 12');
88
```

```
89 •  CREATE TABLE pizza_toppings (
90     topping_id INTEGER,
91     topping_name TEXT
92 );
```



```
95 • INSERT INTO pizza_toppings
96     (topping_id, topping_name)
97 VALUES
98     (1, 'Bacon'),
99     (2, 'BBQ Sauce'),
00     (3, 'Beef'),
01     (4, 'Cheese'),
02     (5, 'Chicken'),
03     (6, 'Mushrooms'),
04     (7, 'Onions'),
05     (8, 'Pepperoni'),
06     (9, 'Peppers'),
07     (10, 'Salami'),
08     (11, 'Tomatoes'),
09     (12, 'Tomato Sauce');
```

```
112
113
114 -- A. Pizza Metrics
115
116 -- How many pizzas were ordered?
117
118 • select count(order_id) as pizza_count from runner_orders;
119
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	pizza_count
▶	10

```
130 -- How many unique customer orders were made?
131
132 • select count(distinct order_id) as unique_customers from runner_orders;
133
134
135
136
137
138
139
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	unique_customers
▶	10

```
142      -- How many successful orders were delivered by each runner?
143 •    SELECT COUNT(*) AS successful_orders
144           ,runner_id FROM runner_orders
145 WHERE cancellation IS NULL
146 GROUP BY runner_id;
147
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	successful_orders	runner_id
▶	3	1
	4	2
	1	3

```
152
153      -- How many of each type of pizza was delivered?
154 •   SELECT COUNT(c.pizza_id) AS numbers_sold
155           ,p.pizza_name
156   FROM customer_orders c
157   JOIN pizza_names p ON c.pizza_id = p.pizza_id
158   GROUP BY p.pizza_name;
159
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	numbers_sold	pizza_name
▶	10	Meatlovers
	4	Vegetarian

```

163  -- How many Vegetarian and Meatlovers were ordered by each customer?
164  •  SELECT COUNT(c.pizza_id) AS numbers_sold
165         ,c.customer_id
166         ,p.pizza_name
167  FROM customer_orders c
168  JOIN pizza_names p ON c.pizza_id = p.pizza_id
169  GROUP BY c.customer_id,p.pizza_name;

```


Result Grid |   Filter Rows: | Export:  | Wrap Cell Content: 

	numbers_sold	customer_id	pizza_name
▶	2	101	Meatlovers
	2	102	Meatlovers
	1	102	Vegetarian
	3	103	Meatlovers
	1	103	Vegetarian
	3	104	Meatlovers
	1	101	Vegetarian
	1	105	Vegetarian

```

175
176  -- What was the maximum number of pizzas delivered in a single order?
177 •  select count(pizza_id), order_id from customer_orders
178      group by order_id
179      having count(pizza_id) >= all
180      (select count(pizza_id) as order_count from customer_orders
181       group by order_id);
182

```

Result Grid |   Filter Rows: | Export:  | Wrap Cell Content: 

	count(pizza_id)	order_id
▶	3	4


```

184      -- For each customer, how many delivered pizzas had at least 1 change and how many had no changes?
185
186 •   select * from customer_orders;
187 •   select customer_id, changes, count(changes) as changes_count from
188   (select *,
189   case when exclusions is not null and extras is not null then 'Y'
190   when exclusions is not null or extras is null then 'N'
191   end as changes
192   from customer_orders) as co
193   group by changes, customer_id;
194
195

```

Result Grid   Filter Rows: | Export:  | Wrap Cell Content: 

	customer_id	changes	changes_count
▶	101	Y	3
	102	Y	2
	102	N	1
	103	Y	4
	104	Y	3
	105	Y	1

```

196 • SELECT customer_id,
197      CASE
198          WHEN exclusions IS NOT NULL OR extras IS NOT NULL THEN 'Y'
199          ELSE 'N'
200      END AS changes,
201      COUNT(*) AS count_changes
202 FROM customer_orders
203 GROUP BY customer_id,
204      CASE
205          WHEN exclusions IS NOT NULL OR extras IS NOT NULL THEN 'Y'
206          ELSE 'N'
207      END;

```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:





	customer_id	changes	count_changes
▶	101	Y	3
	102	Y	3
	103	Y	4
	104	Y	3
	105	Y	1

```

210 • SELECT customer_id,
211     CASE
212         WHEN exclusions IS NOT NULL OR extras IS NOT NULL THEN 'Y'
213         ELSE 'N'
214     END AS changes,
215     COUNT(*) AS count_changes,
216     CASE
217         WHEN exclusions IS NOT NULL OR extras IS NOT NULL THEN 'Y'
218         ELSE 'N'
219     END AS no_changes
220 FROM customer_orders
221 GROUP BY customer_id, changes, no_changes
222 ORDER BY customer_id;
223

```

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

	customer_id	changes	count_changes	no_changes
▶	101	Y	3	Y
	102	Y	3	Y
	103	Y	4	Y
	104	Y	3	Y
	105	Y	1	Y

```

225 • SELECT customer_id,
226         changes,
227         COUNT(*) AS count_changes
228 FROM (
229     SELECT customer_id,
230            CASE
231                WHEN exclusions IS NOT NULL OR extras IS NOT NULL THEN 'Y'
232                ELSE 'N'
233            END AS changes
234     FROM customer_orders
235 ) AS subquery
236 GROUP BY customer_id, changes
237 ORDER BY customer_id, changes;
238

```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	customer_id	changes	count_changes
	101	Y	3
	102	Y	3
	103	Y	4
	104	Y	3
	105	Y	1

```
239 • select * from customer_orders;
240 -- How many pizzas were delivered that had both exclusions and extras?
241 • select count(pizza_id) as Delivered_Pizza from customer_orders
242 where exclusions and extras <> '';
243
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:

	Delivered_Pizza
▶	2

```
248
249 -- What was the total volume of pizzas ordered for each hour of the day?
250 • select hour(order_time) as hours, count(order_id) as number_of_pizzas
251 from customer_orders
252 group by hours;
253
254
255
256
257
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	hours	number_of_pizzas
▶	18	3
	19	1
	23	3
	13	3
	21	3
	11	1


```
261 -- What was the volume of orders for each day of the week?
262 • select dayname(order_time) as days, count(order_id) as no_of_pizzas
263 from customer_orders
264 group by days;
265
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	days	no_of_pizzas
▶	Wednesday	5
	Thursday	3
	Saturday	5
	Friday	1


```
272 -- B. Runner and Customer Experience
273
274 -- How many runners signed up for each 1 week period? (i.e. week starts 2021-01-01)?
275
276 • select
277 week(registration_date) as week_date,
278 count(runner_id) as runner_signed_up from
279 runners
280 group by week_date;
281
282
283
284
```




Result Grid |   Filter Rows: | Export:  | Wrap Cell Content: 

	week_date	runner_signed_up
▶	0	1
	1	2
	2	1

```

286
287 -- What was the average time in minutes it took for each runner to arrive at the Pizza Runner HQ to pickup the order?
288 • select round(avg(timestampdiff(minute,c.order_time,r.pickup_time)),2) as avg_pickup,
289 runner_id from runner_orders r
290 join customer_orders c
291 on r.order_id = c.order_id
292 group by runner_id;

```

Result Grid |   Filter Rows: | Export:  | Wrap Cell Content: 

	avg_pickup	runner_id
▶	15.33	1
	23.40	2
	10.00	3

```

298  -- Is there any relationship between the number of pizzas and how long the order takes to prepare?
299  • with
300  cte as (select round(avg(timestampdiff(minute,c.order_time,r.pickup_time)),2) as avg_pickup,
301          count(c.pizza_id) as pizza_count from runner_orders r
302          join customer_orders c
303          on r.order_id = c.order_id
304          group by c.order_id)
305  select round(avg(avg_pickup),2) , pizza_count from cte
306  group by pizza_count;
307

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	round(avg(avg_pickup),2)	pizza_count
▶	12.00	1
	18.00	2
	29.00	3

```
308
309 -- What was the average distance travelled for each customer?
310 • select customer_id, round(avg(distance),2) as average_distance
311 from runner_orders r
312 join customer_orders c
313 on r.order_id = c.order_id
314 group by customer_id;
315
316
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	customer_id	average_distance
▶	101	13.33
	102	16.73
	103	17.55
	104	10
	105	25

```
320 -- What was the difference between the longest and shortest delivery times for all orders?
321
322
323 • select max(duration+0), min(duration+0),
324        (max(duration+0)- min(duration+0)) as difference
325 from runner_orders
326 where duration+0 != 'NULL';
327
328
329
```

Result Grid   Filter Rows: | Export:  | Wrap Cell Content: 

	max(duration+0)	min(duration+0)	difference
▶	40	10	30