

Tabele:

Employee	
id	int
pesel number	varchar(11)
name	varchar(25)
surname	varchar(25)
sex	Sex enum
birth date	date
city of residence	varchar(35)
street address	varchar(40)
apartment number	varchar(5)
phone number	varchar(12)

Pracownik

Tabela przechowująca dane personalne i kontaktowe poszczególnych pracowników.

ENUM Sex enum:

M

F

Zatrudnienie

Employment	
id	int
job	Jobs enum
hourly wage	decimal(5,2)
date of employment	date
estate id	int

Tabela przechowująca dane związane z zatrudnieniem poszczególnych pracowników.

```
ENUM Jobs enum:  
    manager  
    cook  
    waiter
```

Rozliczenie

Tabela zawierająca zmiany
poszczególnych pracowników.

Settlement	
id	int
employment id	int
shift start	timestamp
shift end	timestamp

Nieruchomość

Tabela zawierająca lokalizację i dane kontaktowe poszczególnych placówek oraz wskazująca pracownika będącego jej menadżerem.

Estate	
id	int
manager	int
city	varchar(35)
street address	varchar(40)
postcode	varchar(6)
phone number	varchar(12)

Stół

Table	
id	int
spots number	int
estate id	int

Tabela przechowująca liczbę miejsc przy danych stołach w poszczególnych placówkach.

Reservation	
id	int
reservation start	timestamp
reservation finished	boolean
table id	int

Rezerwacja

Tabela przechowująca czas początku rezerwacji i stan jej zakończenia oraz wskazująca na rezerwowany stolik.

Menu

Tabela zawierająca informacje dotyczące poszczególnych pozycji z menu.

```
ENUM Category enum:  
    drink  
    appetizer  
    dessert
```

Menu	
id	int
name	varchar(25)
category	Category enum
price	decimal(5,2)
available	boolean

Składnik

Tabela zawierająca składniki potencjalnie niepożądane.

Ingredient	
id	int
name	varchar(25)

Lista składników

Ingredients list	
id	int
menu item id	int
ingredient id	int

Tabela przechowująca listę potencjalnie niepożądanych składników dla danej pozycji z menu.

Rachunek

Bill	
id	int
purchase type	Purchase enum
payment method	Payment enum
purchase time	timestamp

Tabela przechowująca informacje na temat metody płatności, typu zamówienia oraz jego czasu.

```
ENUM Purchase enum:  
    at location  
    takeaway
```

```
ENUM Payment enum:  
    cash  
    card
```

Zamówienie

Tabela zawierająca dane o poszczególnych pozycjach z menu zamówionych na dany rachunek.

Bill order	
id	int
bill id	int
menu item id	int
quantity	int

Zapytania:

Projekcije:

Zwraca czas rezerwacji oraz ich stan

```
SELECT  
    `reservation start`,  
    `reservation finished`  
FROM `Reservation`;
```

	reservation start	reservation finished
	2020-05-01 13:00:00	1
	2020-05-03 10:00:00	1
	2020-05-04 10:30:00	1
	2020-05-04 15:00:00	1
	2020-05-04 15:00:00	1
	2020-05-06 11:30:00	1
	2020-05-07 13:00:00	1
	2020-05-08 12:00:00	1
	2020-05-11 12:30:00	1
	2020-05-11 14:00:00	1
	2020-05-12 13:00:00	1
	2020-05-13 10:30:00	1
	2020-05-13 15:00:00	1
	2020-06-04 11:30:00	1
	2020-06-04 13:00:00	1
	2020-06-05 12:00:00	1
	2020-06-07 10:00:00	1
	2020-06-07 11:00:00	1
	2020-06-07 11:00:00	1
	2020-06-07 13:00:00	1
	2020-06-07 13:30:00	1
	2020-06-08 11:30:00	1
	2020-06-08 14:00:00	1
	2020-06-11 10:00:00	1
	2020-06-12 12:00:00	1
	2020-06-12 14:00:00	1
	2020-06-12 15:00:00	1
	2020-06-11 10:00:00	1
	2020-06-12 12:00:00	1
	2020-06-12 14:00:00	1
	2020-06-12 15:00:00	1
	2020-06-13 12:30:00	1
	2020-06-21 10:00:00	0
	2020-06-27 16:30:00	0

Zwraca podstawowe dane pozycji z menu wraz z jej dostępnością

```
SELECT  
  `name`,  
  `category`,  
  `price`,  
  `available`  
FROM `Menu`;
```

	name	category	price	available
►	Latte	drink	11.40	1
	Americano	drink	11.40	1
	Cappuccino	drink	14.40	0
	Frappe	drink	14.40	1
	Espresso	drink	14.40	1
	Salatka grecka	appetizer	24.20	1
	Salatka nicejska	appetizer	24.20	1
	Salatka Cezar	appetizer	26.20	1
	Salatka Caprese	appetizer	26.20	1
	Salatka Coleslaw	appetizer	22.20	0
	Tiramisu	dessert	16.90	1
	Panna Cotta	dessert	16.90	1
	Crème brûlée	dessert	22.90	1
	Poire belle Hélène	dessert	24.90	1
	Pastéis de Belém	dessert	24.90	1

Zwraca dane osobowe i dane kontaktowe pracowników

```
SELECT
    CONCAT(`name`, ' ', `surname`) AS 'employee',
    `city of residence`,
    `street address`,
    `apartment number`,
    `phone number`
    FROM `Employee`;
```

	employee	city of residence	street address	apartment number	phone number
▶	Zofia Michalik	Kraków	Gołaśka 10	23	+48789070572
	Wiktoria Przybylska	Lublin	Lipowa 10	8	+48792557494
	Jan Czerwiński	Warszawa	Goworowska 6	13	+48608724642
	Stanisław Nowak	Poznań	Nowogrodzka 4	7a	+48885074505
	Wojciech Dudek	Kraków	Stefana Garczyńskiego 19	10	+48509840582
	Małgorzata Górską	Gdańsk	Sochaczewska 16	25	+48791610857
	Patrycja Malinowska	Warszawa	Kurhan 8	5	+48797764730
	Tadeusz Ostrowski	Poznań	Podgórze 10	23b	+48600806538
	Dominika Mazurek	Gdańsk	Michała Glinki 35	8	+48883050649
	Zuzanna Mazur	Gdynia	Kapitańska 2	9c	+48694377631
	Hubert Sawicki	Warszawa	Krypska 15	1	+48533825461
	Krzysztof Jaworski	Lublin	3 Maja 33	14	+48505375408
	Michał Piasecki	Gdańsk	Józefa Kotarbińskiego 6	1	+48604677643
	Oliwier Karpiński	Warszawa	Wytowna 35	7	+48725551266
	Szymon Jarosz	Kraków	Murarska 22	11	+48507984706
	Anna Marciniak	Lublin	Lotnicza 5	2	+48697992577
	Zuzanna Makowska	Warszawa	Adolfa Dygasińskiego 20	12	+48696399861
	Michał Witkowski	Świdnik	Racławicka 13	19	+48506620062
	Hanna Wiśniewska	Kraków	Widna 24	13	+48792169523
	Kacper Kamiński	Warszawa	Antoniego Ponikowskiego 4	2	+48883113942
	Stanisław Pietrzak	Warszawa	Królowej Aldony 18	25	+48729851383
	Gabriela Jaworska	Gdańsk	Strajku Dokerów 27	14	+48667857396
	Wiktor Barański	Warszawa	Pilicka 4	12	+48502075833
	Weronika Wilk	Lublin	Łęczyńska 17	24	+48782737439
	Maciej Żak	Warszawa	Głębzycka 6	20	+48666079458
	Julia Bukowska	Gdynia	Wacława Sieroszewskiego...	14	+48606093341
	Maria Wesołowska	Kraków	Bagrowa 29	23	+48733139605
	Aleksandra Stankiewicz	Poznań	Barzyńskiego 16	23	+48501029363
	Oliwia Wójtowicz	Lublin	Strzelecka 23	14b	+48737519089
	Natalia Woźniak	Kraków	Słupska 26	16	+48666930158

Zwraca dane kontaktowe poszczególnych lokali

```
SELECT
  `city`,
  `street address`,
  `postcode`,
  `phone number`
  FROM `Estate`;
```

	city	street address	postcode	phone number
▶	Lublin	Idziego Radziszewskiego 16	20-031	+48707026624
	Lublin	Krakowskie Przedmieście 4	20-002	+48698253087
	Warszawa	plac Thomasa Woodrowa Wilsona 2	01-615	+48764230974
	Kraków	plac Szczepański 2	31-011	+48750945082
	Poznań	Krysiewicza 6	61-825	+48936501845
	Kraków	Rakowicka 17	31-511	+48942375933
	Gdańsk	Szafarnia 6	80-755	+48572405435
	Warszawa	Żelazna 32	00-832	+48567249614
	Warszawa	aleja Wyzwolenia 18	00-570	+48593647246
	Sopot	Powstańców Warszawy 10	81-718	+48924724073

Zwraca średnią cenę pozycji z menu względem jej kategorii

```
SELECT
  `category`,
  ROUND(AVG(`price`), 2)
FROM `Menu`
GROUP BY `category`
ORDER BY `category`;
```

	category	price
▶	drink	13.20
	appetizer	24.60
	dessert	21.30

Zwraca średnią stawkę godzinową w zależności od wykonywanego zawodu

```
SELECT
  `job`,
  ROUND(AVG(`hourly wage`), 2) AS 'hourly wage'
FROM `Employment`
GROUP BY `job`
ORDER BY `job`;
```

	job	hourly wage
▶	manager	53.10
	cook	29.10
	waiter	13.30

Zwraca ilość stolików ze względu na ilość miejsc

```
SELECT
  `spots number`,
  COUNT(id) AS 'tables number'
FROM `Table`
GROUP BY `spots number`;
```

	spots number	tables number
▶	2	10
	3	10
	4	10

Zwraca ilość zamówień ze względu na ich typ

```
SELECT
  `purchase type`,
  COUNT(`id`) AS 'quantity'
  FROM `Bill`
  GROUP BY `purchase type`
  ORDER BY `purchase type`;
```

	purchase type	quantity
▶	at location	11
	takeaway	4

Zwraca ilość zamówień ze względu na typ płatności

```
SELECT
    `payment method`,
    COUNT(`id`) AS 'quantity'
FROM `Bill`
GROUP BY `payment method`
ORDER BY `payment method`;
```

	payment method	quantity
▶	cash	6
	card	9

Zwraca średnią ilość pozycji z menu na jedno zamówienie

```
SELECT  
    ROUND(SUM(`quantity`) / COUNT(DISTINCT `bill id`), 2) AS 'orders'  
FROM `Bill order`;
```

	orders
▶	2.53

Selekcje:

Zwraca rachunki zamówień
na miejscu

```
SELECT  
  *  
  FROM `Bill`  
  WHERE `purchase type` = 'at location';
```

	id	purchase type	payment method	purchase time
▶	1	at location	card	2020-05-04 12:30:16
	2	at location	cash	2020-05-04 12:54:18
	3	at location	cash	2020-05-05 10:57:45
	4	at location	card	2020-05-08 12:02:10
	6	at location	card	2020-06-02 15:25:03
	7	at location	card	2020-06-03 16:17:55
	8	at location	card	2020-06-03 17:08:21
	9	at location	cash	2020-06-04 14:46:10
	11	at location	cash	2020-06-05 15:23:58
	14	at location	card	2020-06-12 11:44:52
	15	at location	cash	2020-06-13 17:17:53
✱	NULL	NULL	NULL	NULL

Zwraca rachunki zamówień na wynos

```
SELECT
    *
    FROM `Bill`
    WHERE `purchase type` = 'takeaway';
```

	id	purchase type	payment method	purchase time
▶	5	takeaway	cash	2020-05-12 10:39:18
	10	takeaway	card	2020-06-05 11:45:03
	12	takeaway	card	2020-06-06 15:38:10
	13	takeaway	card	2020-06-08 11:38:45
✱	NULL	NULL	NULL	NULL

Zwraca rachunki zamówień opłaconych kartą

```
SELECT
    *
FROM `Bill`
WHERE `payment method` = 'card';
```

	id	purchase type	payment method	purchase time
▶	1	at location	card	2020-05-04 12:30:16
	4	at location	card	2020-05-08 12:02:10
	6	at location	card	2020-06-02 15:25:03
	7	at location	card	2020-06-03 16:17:55
	8	at location	card	2020-06-03 17:08:21
	10	takeaway	card	2020-06-05 11:45:03
	12	takeaway	card	2020-06-06 15:38:10
	13	takeaway	card	2020-06-08 11:38:45
	14	at location	card	2020-06-12 11:44:52
✱	NULL	NULL	NULL	NULL

Zwraca rachunki zamówień opłaconych gotówką

```
SELECT
    *
FROM `Bill`
WHERE `payment method` = 'cash';
```

	id	purchase type	payment method	purchase time
▶	2	at location	cash	2020-05-04 12:54:18
	3	at location	cash	2020-05-05 10:57:45
	5	takeaway	cash	2020-05-12 10:39:18
	9	at location	cash	2020-06-04 14:46:10
	11	at location	cash	2020-06-05 15:23:58
	15	at location	cash	2020-06-13 17:17:53
✱	NULL	NULL	NULL	NULL

Zwraca dostępne
pozycje z menu

```
SELECT  
  *  
FROM `Menu`  
WHERE `available` = true;
```

	id	name	category	price	available
▶	1	Latte	drink	11.40	1
	2	Americano	drink	11.40	1
	4	Frappe	drink	14.40	1
	5	Espresso	drink	14.40	1
	6	Salatka grecka	appetizer	24.20	1
	7	Salatka nicejska	appetizer	24.20	1
	8	Salatka Cezar	appetizer	26.20	1
	9	Salatka Caprese	appetizer	26.20	1
	11	Tiramisu	dessert	16.90	1
	12	Panna Cotta	dessert	16.90	1
	13	Crème brûlée	dessert	22.90	1
	14	Poire belle Hélène	dessert	24.90	1
	15	Pastéis de Belém	dessert	24.90	1
✱	NULL	NULL	NULL	NULL	NULL

Zwraca niedostępne pozycje z menu

```
SELECT
    *
    FROM `Menu`
    WHERE `available` = false;
```

	id	name	category	price	available
▶	3	Cappuccino	drink	14.40	0
	10	Salatka Coleslaw	appetizer	22.20	0
✱	NULL	NULL	NULL	NULL	NULL

Zwraca napoje z menu

```
SELECT
  *
  FROM `Menu`
 WHERE `category` = 'drink';
```

	id	name	category	price	available
▶	1	Latte	drink	11.40	1
	2	Americano	drink	11.40	1
	3	Cappuccino	drink	14.40	0
	4	Frappe	drink	14.40	1
	5	Espresso	drink	14.40	1
✱	NULL	NULL	NULL	NULL	NULL

Zwraca przystawki z menu

```
SELECT
    *
FROM `Menu`
WHERE `category` = 'appetizer';
```

	id	name	category	price	available
▶	6	Salatka grecka	appetizer	24.20	1
	7	Salatka nicejska	appetizer	24.20	1
	8	Salatka Cezar	appetizer	26.20	1
	9	Salatka Caprese	appetizer	26.20	1
	10	Salatka Coleslaw	appetizer	22.20	0
✱	NULL	NULL	NULL	NULL	NULL

Zwraca desery z menu

```
SELECT
    *
FROM `Menu`
WHERE `category` = 'dessert';
```

	id	name	category	price	available
▶	11	Tiramisu	dessert	16.90	1
	12	Panna Cotta	dessert	16.90	1
	13	Crème brûlée	dessert	22.90	1
	14	Poire belle Hélène	dessert	24.90	1
	15	Pastéis de Belém	dessert	24.90	1
✱	NULL	NULL	NULL	NULL	NULL

Zwraca niezakończone rezerwacje

```
SELECT
    *
FROM `Reservation`
WHERE `reservation finished` = false;
```

	id	reservation start	reservation finished	table id
▶	29	2020-06-21 10:00:00	0	10
	30	2020-06-27 16:30:00	0	19
•	NULL	NULL	NULL	NULL

Zapytania łączące dwie tabele:

Zwraca wartości rachunków

```
SELECT
  SUM(`Menu`.`price` * `Bill order`.`quantity`) AS 'price'
FROM (`Bill order`
      INNER JOIN `Menu` ON `Bill order`.`menu item id` = `Menu`.`id`)
GROUP BY `Bill order`.`bill id`
ORDER BY SUM(`Menu`.`price` * `Bill order`.`quantity`) DESC;
```

	price
▶	106.40
	105.90
	99.60
	78.60
	77.20
	65.50
	61.20
	51.10
	48.40
	22.90
	22.80
	14.40
	11.40
	11.40
	11.40

Zwraca ilość zamówień danych napojów

```
SELECT
  `Menu`.`name`,
  SUM(`Bill order`.`quantity`) AS 'quantity'
FROM (`Bill order`
      INNER JOIN `Menu` ON `Bill order`.`menu item id` = `Menu`.`id`)
WHERE `Menu`.`category` = 'drink'
GROUP BY `Menu`.`name`
ORDER BY SUM(`Bill order`.`quantity`) DESC;
```

	name	quantity
▶	Latte	4
	Espresso	4
	Americano	2
	Frappe	1

Zwraca ilość zamówień danych przystawek

```
SELECT
  `Menu`.`name`,
  SUM(`Bill order`.`quantity`) AS 'quantity'
FROM (`Bill order`
      INNER JOIN `Menu` ON `Bill order`.`menu item id` = `Menu`.`id`)
WHERE `Menu`.`category` = 'appetizer'
GROUP BY `Menu`.`name`
ORDER BY SUM(`Bill order`.`quantity`) DESC;
```

	name	quantity
►	Salatka grecka	5
	Salatka Cezar	5
	Salatka nicejska	4
	Salatka Caprese	1

Zwraca ilość zamówień danych deserów

```
SELECT
  `Menu`.`name`,
  SUM(`Bill order`.`quantity`) AS 'quantity'
FROM (`Bill order`
      INNER JOIN `Menu` ON `Bill order`.`menu item id` = `Menu`.`id`)
WHERE `Menu`.`category` = 'dessert'
GROUP BY `Menu`.`name`
ORDER BY SUM(`Bill order`.`quantity`) DESC;
```

	name	quantity
►	Poire belle Hélène	5
	Pastéis de Belém	3
	Tiramisu	2
	Panna Cotta	1
	Crème brûlée	1

Zwraca ilość zamówień z zeszłego miesiąca ze względu na typ zamówienia

```
SELECT
  `Bill`.`purchase type`,
  SUM(`Bill order`.`quantity`) AS 'quantity'
FROM (`Bill order`
      INNER JOIN `Bill` ON `Bill order`.`bill id` = `Bill`.`id`)
WHERE MONTH(`Bill`.`purchase time`) = MONTH(CURRENT_DATE()) - 1
GROUP BY `Bill`.`purchase type`
ORDER BY SUM(`Bill order`.`quantity`) DESC;
```

	purchase type	quantity
▶	at location	11
	takeaway	2

Zwraca ilość zamówień z zeszłego miesiąca ze
względu na metodę płatności

```
SELECT
  `Bill`.`payment method`,
  SUM(`Bill order`.`quantity`) AS 'quantity'
FROM (`Bill order`
      INNER JOIN `Bill` ON `Bill order`.`bill id` = `Bill`.`id`)
WHERE MONTH(`Bill`.`purchase time`) = MONTH(CURRENT_DATE()) - 1
GROUP BY `Bill`.`payment method`
ORDER BY SUM(`Bill order`.`quantity`) DESC;
```

	payment method	quantity
►	cash	8
	card	5

Zwraca wartość wynagrodzeń należnych za poprzedni miesiąc ze względu na wykonywany zawód

```
SELECT
  `Employment`.`job`,
  ROUND(SUM((TIMESTAMPDIFF(SECOND, `Settlement`.`shift start`, `Settlement`.`shift end`) / 3600) * `Employment`.`hourly wage`), 2) as 'payment'
FROM (`Employment`
      INNER JOIN `Settlement` ON `Employment`.`id` = `Settlement`.`employment id`)
WHERE MONTH(`Settlement`.`shift start`) = MONTH(CURRENT_DATE()) - 1
GROUP BY `Employment`.`job`
ORDER BY ROUND(SUM((TIMESTAMPDIFF(SECOND, `Settlement`.`shift start`, `Settlement`.`shift end`) / 3600) * `Employment`.`hourly wage`), 2) DESC;
```

	job	payment
►	manager	4426.87
	cook	3198.07
	waiter	1527.19

Zwraca ilość godzin przepracowanych w poprzednim miesiącu ze względu na wykonywany zawód

```
SELECT
  `Employment`.`job`,
  SUM(TIMESTAMPDIFF(SECOND, `Settlement`.`shift start`, `Settlement`.`shift end`) / 3600) AS `hours`
FROM (`Employment`
      INNER JOIN `Settlement` ON `Employment`.`id` = `Settlement`.`employment id`)
WHERE MONTH(`Settlement`.`shift start`) = MONTH(CURRENT_DATE()) - 1
GROUP BY `Employment`.`job`
ORDER BY SUM(TIMESTAMPDIFF(SECOND, `Settlement`.`shift start`, `Settlement`.`shift end`) / 3600) DESC;
```

	job	hours
▶	cook	108.7035
	waiter	108.2707
	manager	84.2639

Zwraca liczbę rezerwacji z poprzedniego miesiąca
ze względu na ilość miejsc

```
SELECT
  `Table`.`spots number`,
  COUNT(`Reservation`.`id`) AS 'quantity'
FROM (`Table`
      INNER JOIN `Reservation` ON `Table`.`id` = `Reservation`.`table id`)
WHERE MONTH(`Reservation`.`reservation start`) = MONTH(CURRENT_DATE()) - 1
GROUP BY `Table`.`spots number`
ORDER BY COUNT(`Reservation`.`id`) DESC;
```

	spots number	quantity
►	2	5
	3	5
	4	3

Zwraca liczbę rezerwacji z poprzedniego miesiąca ze względu na lokal

	estate id	quantity
▶	2	3
	10	3
	4	2
	8	2
	3	1
	5	1
	6	1

```
SELECT
  `Table`.`estate id`,
  COUNT(`Reservation`.`id`) AS 'quantity'
FROM (`Table`
      INNER JOIN `Reservation` ON `Table`.`id` = `Reservation`.`table id`)
WHERE MONTH(`Reservation`.`reservation start`) = MONTH(CURRENT_DATE()) - 1
GROUP BY `Table`.`estate id`
ORDER BY COUNT(`Reservation`.`id`) DESC;
```

Zapytania łączące trzy tabele:

Zwraca wartości rachunków z
zeszłego miesiąca

	price
▶	106.40
	99.60
	22.80
	11.40
	11.40

```
SELECT
    SUM(`Menu`.`price` * `Bill order`.`quantity`) AS 'price'
    FROM ((`Bill order`
        INNER JOIN `Menu` ON `Bill order`.`menu item id` = `Menu`.`id`)
        INNER JOIN `Bill` ON `Bill order`.`bill id` = `Bill`.`id`)
        WHERE MONTH(`Bill`.`purchase time`) = MONTH(CURRENT_DATE()) - 1
        GROUP BY `Bill`.`id`
        ORDER BY SUM(`Menu`.`price` * `Bill order`.`quantity`) DESC;
```


Zwraca wartości rachunków z zeszłego miesiąca ze
względu na typ zamówienia

```
SELECT
  `Bill`.`purchase type`,
  SUM(`Menu`.`price` * `Bill order`.`quantity`) AS 'price'
FROM ((`Bill order`
      INNER JOIN `Menu` ON `Bill order`.`menu item id` = `Menu`.`id`)
      INNER JOIN `Bill` ON `Bill order`.`bill id` = `Bill`.`id`)
WHERE MONTH(`Bill`.`purchase time`) = MONTH(CURRENT_DATE()) - 1
GROUP BY `Bill`.`purchase type`
ORDER BY SUM(`Menu`.`price` * `Bill order`.`quantity`) DESC;
```

	purchase type	price
▶	at location	228.80
	takeaway	22.80

Zwraca wartości rachunków z zeszłego miesiąca ze
względu na metodę płatności

```
SELECT
  `Bill`.`payment method`,
  SUM(`Menu`.`price` * `Bill order`.`quantity`) AS 'price'
FROM ((`Bill order`
  INNER JOIN `Menu` ON `Bill order`.`menu item id` = `Menu`.`id`)
  INNER JOIN `Bill` ON `Bill order`.`bill id` = `Bill`.`id`)
WHERE MONTH(`Bill`.`purchase time`) = MONTH(CURRENT_DATE()) - 1
GROUP BY `Bill`.`payment method`
ORDER BY SUM(`Menu`.`price` * `Bill order`.`quantity`) DESC;
```

	payment method	price
▶	cash	140.60
	card	111.00

Zwraca wartość zamówień z zeszłego miesiąca ze względu na ich kategorię

```
SELECT
  `Menu`.`category`,
  SUM(`Menu`.`price`) as 'price'
FROM ((`Bill order`
      INNER JOIN `Menu` ON `Bill order`.`menu item id` = `Menu`.`id`)
      INNER JOIN `Bill` ON `Bill order`.`bill id` = `Bill`.`id`)
WHERE MONTH(`Bill`.`purchase time`) = MONTH(CURRENT_DATE()) - 1
GROUP BY `Menu`.`category`
ORDER BY SUM(`Menu`.`price`) DESC;
```

	category	price
▶	dessert	66.70
	drink	34.20
	appetizer	24.20

Zwraca ilości zamówień z zeszłego miesiąca ze względu na ich kategorię

```
SELECT
  `Menu`.`category`,
  SUM(`Bill order`.`quantity`) AS 'quantity'
FROM ((`Bill order`
      INNER JOIN `Menu` ON `Bill order`.`menu item id` = `Menu`.`id`)
      INNER JOIN `Bill` ON `Bill order`.`bill id` = `Bill`.`id`)
WHERE MONTH(`Bill`.`purchase time`) = MONTH(CURRENT_DATE()) - 1
GROUP BY `Menu`.`category`
ORDER BY SUM(`Bill order`.`quantity`) DESC;
```

	category	quantity
▶	dessert	6
	drink	4
	appetizer	3

Zwraca ilość danych napojów zamówionych w poprzednim miesiącu

```
SELECT
  `Menu`.`name`,
  SUM(`Bill order`.`quantity`) AS 'quantity'
FROM ((`Bill order`
  INNER JOIN `Menu` ON `Bill order`.`menu item id` = `Menu`.`id`)
  INNER JOIN `Bill` ON `Bill order`.`bill id` = `Bill`.`id`)
  WHERE `Menu`.`category` = 'drink' AND MONTH(`Bill`.`purchase time`) = MONTH(CURRENT_DATE()) - 1
  GROUP BY `Menu`.`name`
  ORDER BY SUM(`Bill order`.`quantity`) DESC;
```

	name	quantity
▶	Latte	3
	Americano	1

Zwraca ilość danych przystawek zamówionych w poprzednim miesiącu

```
SELECT
  `Menu`.`name`,
  SUM(`Bill order`.`quantity`) AS 'quantity'
FROM ((`Bill order`
  INNER JOIN `Menu` ON `Bill order`.`menu item id` = `Menu`.`id`)
  INNER JOIN `Bill` ON `Bill order`.`bill id` = `Bill`.`id`)
  WHERE `Menu`.`category` = 'appetizer' AND MONTH(`Bill`.`purchase time`) = MONTH(CURRENT_DATE()) - 1
  GROUP BY `Menu`.`name`
  ORDER BY SUM(`Bill order`.`quantity`) DESC;
```

	name	quantity
▶	Salatka grecka	3

Zwraca ilość danych deserów zamówionych w poprzednim miesiącu

```
SELECT
  `Menu`.`name`,
  SUM(`Bill order`.`quantity`) AS 'quantity'
FROM ((`Bill order`
  INNER JOIN `Menu` ON `Bill order`.`menu item id` = `Menu`.`id`)
  INNER JOIN `Bill` ON `Bill order`.`bill id` = `Bill`.`id`)
WHERE `Menu`.`category` = 'dessert' AND MONTH(`Bill`.`purchase time`) = MONTH(CURRENT_DATE()) - 1
GROUP BY `Menu`.`name`
ORDER BY SUM(`Bill order`.`quantity`) DESC;
```

	name	quantity
▶	Tiramisu	2
	Poire belle Hélène	2
	Pastéis de Belém	2

Zwraca dane osobowe, wykonywany zawód
oraz ilość godzin przepracowanych w zeszłym
miesiącu

```
SELECT
  CONCAT(`Employee`.`name`, ' ', `Employee`.`surname`) AS 'employee',
  `Employment`.`job`,
  SUM(TIMESTAMPDIFF(SECOND, `Settlement`.`shift start`, `Settlement`.`shift end`) / 3600) AS `hours`
FROM ((`Employment`
  INNER JOIN `Settlement` ON `Employment`.`id` = `Settlement`.`employment id`)
  INNER JOIN `Employee` ON `Employment`.`id` = `Employee`.`id`)
  WHERE MONTH(`Settlement`.`shift start`) = MONTH(CURRENT_DATE()) - 1
  GROUP BY `Employee`.`id`
  ORDER BY SUM(TIMESTAMPDIFF(SECOND, `Settlement`.`shift start`, `Settlement`.`shift end`) / 3600) DESC;
```

	employee	job	hours
►	Oliwia Wójtowicz	manager	19.1778
	Hubert Sawicki	waiter	18.9528
	Szymon Jarosz	cook	18.9108
	Dominika Mazurek	waiter	18.9078
	Wiktor Barański	manager	18.8678
	Wiktoria Przybylska	manager	18.8228
	Natalia Woźniak	waiter	18.7803
	Zuzanna Mazur	cook	16.9550
	Stanisław Nowak	waiter	16.9489
	Gabriela Jaworska	waiter	16.8406
	Jan Czerwiński	cook	9.5344
	Małgorzata Górka	manager	9.5222
	Weronika Wilk	cook	9.4678
	Tadeusz Ostrowski	cook	9.4633
	Stanisław Pietrzak	cook	9.4622
	Julia Bukowska	manager	9.4511
	Michał Witkowski	waiter	9.4050
	Krystian Jaworski	cook	9.3889
	Patrycja Malinowska	cook	8.5811
	Zofia Michalik	cook	8.5433
	Maciej Żak	waiter	8.4353
	Aleksandra Stanki...	manager	8.4222
	Michał Piasecki	cook	8.3967

Zwraca dane osobowe, stawkę godzinową, ilość godzin przepracowanych w zeszłym miesiącu oraz wynagrodzenie należne za poprzedni miesiąc

	employee	hourly wage	hours	payment
▶	Wiktoria Przybylska	51.00	18.8228	959.96
	Małgorzata Górská	54.00	9.5222	514.20
	Wiktor Barański	52.00	18.8678	981.12
	Julia Bukowska	57.00	9.4511	538.71
	Aleksandra Stankiewicz	54.00	8.4222	454.80
	Oliwia Wójtowicz	51.00	19.1778	978.07
	Zofia Michalik	32.00	8.5433	273.39
	Jan Czerwiński	25.00	9.5344	238.36
	Patrycja Malinowska	33.00	8.5811	283.18
	Tadeusz Ostrowski	26.00	9.4633	246.05
	Zuzanna Mazur	33.00	16.9550	559.52
	Krystian Jaworski	27.00	9.3889	253.50
	Michał Piasecki	26.00	8.3967	218.31
	Szymon Jarosz	30.00	18.9108	567.33
	Stanisław Pietrzak	28.00	9.4622	264.94
	Weronika Wilk	31.00	9.4678	293.50
	Stanisław Nowak	11.00	16.9489	186.44
	Dominika Mazurek	17.00	18.9078	321.43
	Hubert Sawicki	14.00	18.9528	265.34
	Michał Witkowski	11.00	9.4050	103.46
	Gabriela Jaworska	14.00	16.8406	235.77
	Maciej Żak	18.00	8.4353	151.84
	Natalia Woźniak	14.00	18.7803	262.92

```
SELECT
  CONCAT(`Employee`.`name`, ' ', `Employee`.`surname`) AS 'employee',
  `Employment`.`hourly wage`,
  SUM(TIMESTAMPDIFF(SECOND, `Settlement`.`shift start`, `Settlement`.`shift end`) / 3600) AS `hours`,
  ROUND(SUM((TIMESTAMPDIFF(SECOND, `Settlement`.`shift start`, `Settlement`.`shift end`) / 3600) * `Employment`.`hourly wage`), 2) as 'payment'
FROM (`Employment`
      INNER JOIN `Settlement` ON `Employment`.`id` = `Settlement`.`employment id`)
      INNER JOIN `Employee` ON `Employment`.`id` = `Employee`.`id`)
WHERE MONTH(`Settlement`.`shift start`) = MONTH(CURRENT_DATE()) - 1
GROUP BY `Employee`.`id`
ORDER BY `Employment`.`job`;
```