

```
SELECT *
FROM `modulabs_project.data`
LIMIT 10;
```

행	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice
1	536365	85123A	WHITE HANGING HEART T-LIG...	6	2010-12-01 08:26:00 UTC	2.55
2	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00 UTC	3.39
3	536365	844068	CREAM CUPID HEARTS COAT H...	8	2010-12-01 08:26:00 UTC	2.75
4	536365	84029G	KNITTED UNION FLAG HOT WA...	6	2010-12-01 08:26:00 UTC	3.39
5	536365	84029E	RED WOOLLY HOTTIE WHITE H...	6	2010-12-01 08:26:00 UTC	3.39
6	536365	22752	SET 7 BABUSHKA NESTING BO...	2	2010-12-01 08:26:00 UTC	7.65
7	536365	21730	GLASS STAR FROSTED T-LIGHT...	6	2010-12-01 08:26:00 UTC	4.25
8	536366	22633	HAND WARMER UNION JACK	6	2010-12-01 08:28:00 UTC	1.85
9	536366	22632	HAND WARMER RED POLKA DOT	6	2010-12-01 08:28:00 UTC	1.85
10	536367	84879	ASSORTED COLOUR BIRD ORN...	32	2010-12-01 08:34:00 UTC	1.69

```
SELECT COUNT(*)
FROM `modulabs_project.data`;
```

행	f0_
1	541909

```
SELECT
  COUNT(InvoiceNo) AS InvoiceNo,
  COUNT(StockCode) AS StockCode,
  COUNT(Description) AS Description,
  COUNT(Quantity) AS Quantity,
  COUNT(InvoiceDate) AS InvoiceDate,
  COUNT(UnitPrice) AS UnitPrice,
  COUNT(CustomerID) AS CustomerID,
  COUNT(Country) AS Country
FROM `modulabs_project.data`;
```

행	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
1	541909	541909	540455	541909	541909	541909	406829	541909

아래는 CASE WHEN 을 사용하여 InvoiceNo의 결측치 비율을 계산하는 쿼리문 예시입니다.

예시

```
SELECT
  'InvoiceNo' AS column_name,
  ROUND(SUM(CASE WHEN InvoiceNo IS NULL THEN 1 ELSE 0 END) / COUNT(*)
  * 100, 2) AS missing_percentage
FROM modulabs_project.data;
```

행	column_name ▼	missing_percenta...
1	InvoiceNo	0.0

해당 쿼리문을 다른 컬럼에도 동일하게 반영한 후, UNION ALL로 연결해 보세요.

-- 각 컬럼의 결측치 비율 계산 쿼리

SELECT

```
'InvoiceNo' AS column_name,
ROUND(SUM(CASE WHEN InvoiceNo IS NULL THEN 1 ELSE 0 END) / COUNT(*)
* 100, 2) AS missing_percentage
FROM `modulabs_project.data`
```

UNION ALL

SELECT

```
'StockCode' AS column_name,
ROUND(SUM(CASE WHEN StockCode IS NULL THEN 1 ELSE 0 END) / COUNT(*)
* 100, 2) AS missing_percentage
FROM `modulabs_project.data`
```

UNION ALL

SELECT

```
'Description' AS column_name,
ROUND(SUM(CASE WHEN Description IS NULL THEN 1 ELSE 0 END) /
COUNT(*) * 100, 2) AS missing_percentage
FROM `modulabs_project.data`
```

UNION ALL

SELECT

```
'Quantity' AS column_name,
ROUND(SUM(CASE WHEN Quantity IS NULL THEN 1 ELSE 0 END) / COUNT(*) *
100, 2) AS missing_percentage
FROM `modulabs_project.data`
```

UNION ALL

SELECT

```

        'InvoiceDate' AS column_name,
        ROUND(SUM(CASE WHEN InvoiceDate IS NULL THEN 1 ELSE 0 END) /
COUNT(*) * 100, 2) AS missing_percentage
FROM `modulabs_project.data`

```

UNION ALL

```

SELECT
        'UnitPrice' AS column_name,
        ROUND(SUM(CASE WHEN UnitPrice IS NULL THEN 1 ELSE 0 END) / COUNT(*)
* 100, 2) AS missing_percentage
FROM `modulabs_project.data`

```

UNION ALL

```

SELECT
        'CustomerID' AS column_name,
        ROUND(SUM(CASE WHEN CustomerID IS NULL THEN 1 ELSE 0 END) /
COUNT(*) * 100, 2) AS missing_percentage
FROM `modulabs_project.data`

```

UNION ALL

```

SELECT
        'Country' AS column_name,
        ROUND(SUM(CASE WHEN Country IS NULL THEN 1 ELSE 0 END) / COUNT(*) *
100, 2) AS missing_percentage
FROM `modulabs_project.data`;

```

행	column_name ▼	missing_percenta...
1	CustomerID	24.93
2	Country	0.0
3	InvoiceDate	0.0
4	Quantity	0.0
5	UnitPrice	0.0
6	Description	0.27
7	StockCode	0.0
8	InvoiceNo	0.0

StockCode = '85123A'의 Description을 추출하는 쿼리문을 작성해 보세요.

```
SELECT DISTINCT Description
```

```
FROM `modulabs_project.data`
```

```
WHERE StockCode = '85123A';
```

행	Description ▼
1	WHITE HANGING HEART T-LIG...
2	?
3	wrongly marked carton 22804
4	CREAM HANGING HEART T-LIG...

```
DELETE FROM `modulabs_project.data`
```

```
WHERE CustomerID IS NULL OR Description IS NULL;
```

i 이 문으로 data의 행 135,080개가 삭제되었습니다.

```
SELECT
```

```
InvoiceNo, StockCode, Description, Quantity, InvoiceDate, UnitPrice, CustomerID,
Country,
```

```
COUNT(*) AS duplicate_count
```


```
FROM `modulabs_project.data`
```

```
GROUP BY InvoiceNo, StockCode, Description, Quantity, InvoiceDate, UnitPrice,
CustomerID, Country
```

```
HAVING COUNT(*) > 1;
```

행	InvoiceNo ▼	StockCode ▼	Description ▼	Quantity ▼	InvoiceDate ▼	UnitPrice ▼
1	571034	23239	SET OF 4 KNICK KNACK TINS P...	6	2011-10-13 12:47:00 UTC	4.15
2	571034	23245	SET OF 3 REGENCY CAKE TINS	4	2011-10-13 12:47:00 UTC	4.95
3	571034	23494	VINTAGE DOLLY DELUXE SEWIN...	3	2011-10-13 12:47:00 UTC	5.95
4	538826	22749	FELTCRAFT PRINCESS CHARLO...	1	2010-12-14 12:58:00 UTC	3.75
5	577228	23048	SET OF 10 LANTERNS FAIRY LL...	1	2011-11-18 12:07:00 UTC	4.15
6	577228	84580	MOUSE TOY WITH PINK T-SHIRT	1	2011-11-18 12:07:00 UTC	3.75
7	577228	22144	CHRISTMAS CRAFT LITTLE FRI...	1	2011-11-18 12:07:00 UTC	2.1
8	577228	22435	SET OF 9 HEART SHAPED BALL...	1	2011-11-18 12:07:00 UTC	1.25
9	577228	23156	SET OF 5 MINI GROCERY MAG...	1	2011-11-18 12:07:00 UTC	2.08
10	577228	22270	HAPPY EASTER HANGING DEC...	1	2011-11-18 12:07:00 UTC	3.75
11	539419	48138	DOORMAT UNION FLAG	10	2010-12-17 14:10:00 UTC	6.75
12	538174	22326	ROUND SNACK BOXES SET OF4...	12	2010-12-10 09:35:00 UTC	2.95
13	555162	22704	WRAP RED APPLES	25	2011-06-01 10:15:00 UTC	0.42
14	576607	22086	PAPER CHAIN KIT 50'S CHRIST...	6	2011-11-15 15:42:00 UTC	2.95
15	576607	22940	FELTCRAFT CHRISTMAS FAIRY	8	2011-11-15 15:42:00 UTC	4.25
16	553731	23208	LUNCH BAG VINTAGE LEAF DE...	10	2011-05-19 08:26:00 UTC	1.65

```
CREATE OR REPLACE TABLE `modulabs_project.data` AS
SELECT DISTINCT *
FROM `modulabs_project.data`;
```

 이 문으로 이름이 data인 테이블이 교체되었습니다.

```
SELECT COUNT(DISTINCT InvoiceNo) AS unique_invoice_count
FROM `modulabs_project.data`;
```

행	unique_invoice_c...
1	22190

```
SELECT DISTINCT InvoiceNo
FROM `modulabs_project.data`
LIMIT 100;
```

행	InvoiceNo
1	541431
2	C541433
3	537626
4	542237
5	549222
6	556201
7	562032
8	573511
9	581180
10	539318
11	541998
12	548955
13	568172
14	577609
15	543037
16	544156

페이지당 결과 수: 50 1 - 50 (전체 100행) |< < > >|

```
SELECT *
FROM `modulabs_project.data`
WHERE InvoiceNo LIKE 'C%'
LIMIT 100;
```

행	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice
1	C541433	23166	MEDIUM CERAMIC TOP STORA...	-74215	2011-01-18 10:17:00 UTC	1.04
2	C545329	M	Manual	-1	2011-03-01 15:47:00 UTC	183.75
3	C545329	M	Manual	-1	2011-03-01 15:47:00 UTC	280.05
4	C545330	M	Manual	-1	2011-03-01 15:49:00 UTC	376.5
5	C547388	21914	BLUE HARMONICA IN BOX	-12	2011-03-22 16:07:00 UTC	1.25
6	C547388	22413	METAL SIGN TAKE IT OR LEAVE...	-6	2011-03-22 16:07:00 UTC	2.95
7	C547388	22701	PINK DOG BOWL	-6	2011-03-22 16:07:00 UTC	2.95
8	C547388	22645	CERAMIC HEART FAIRY CAKE ...	-12	2011-03-22 16:07:00 UTC	1.45
9	C547388	22784	LANTERN CREAM GAZEBO	-3	2011-03-22 16:07:00 UTC	4.95
10	C547388	37448	CERAMIC CAKE DESIGN SPOTT...	-12	2011-03-22 16:07:00 UTC	1.49
11	C547388	84050	PINK HEART SHAPE EGG FRYIN...	-12	2011-03-22 16:07:00 UTC	1.65
12	C549955	22839	3 TIER CAKE TIN GREEN AND C...	-2	2011-04-13 13:38:00 UTC	14.95
13	C549955	22666	RECIPE BOX PANTRY YELLOW ...	-2	2011-04-13 13:38:00 UTC	2.95
14	C580165	23245	SET OF 3 REGENCY CAKE TINS	-2	2011-12-02 11:21:00 UTC	4.95
15	C580165	22720	SET OF 3 CAKE TINS PANTRY D...	-1	2011-12-02 11:21:00 UTC	4.95
16	C580165	22826	LOVE SEAT ANTIQUE WHITE M...	-1	2011-12-02 11:21:00 UTC	42.5

페이지당 결과 수: 50 1 - 50 (전체 100행) |< < > >|

```
SELECT
ROUND(
SUM(CASE WHEN InvoiceNo LIKE 'C%' THEN 1 ELSE 0 END)
/ COUNT(*) * 100,
1) AS canceled_percentage
FROM `modulabs_project.data`;
```

행	canceled_percent...
1	2.2

```
SELECT COUNT(DISTINCT StockCode) AS unique_stockcode_count
FROM `modulabs_project.data`;
```

행	unique_stockcod...
1	3684

```

SELECT StockCode, COUNT(*) AS sell_cnt
FROM `modulabs_project.data`
GROUP BY StockCode
ORDER BY sell_cnt DESC
LIMIT 10;

```

행	StockCode ▼	sell_cnt ▼
1	85123A	2065
2	22423	1894
3	85099B	1659
4	47566	1409
5	84879	1405
6	20725	1346
7	22720	1224
8	POST	1196
9	22197	1110
10	23203	1108

```

WITH UniqueStockCodes AS (
  SELECT DISTINCT StockCode
  FROM `modulabs_project.data`
)
SELECT
  LENGTH(StockCode) - LENGTH(REGEXP_REPLACE(StockCode, r'[0-9]', '')) AS
number_count,
  COUNT(*) AS stock_cnt
FROM UniqueStockCodes
GROUP BY number_count
ORDER BY stock_cnt DESC;

```

행	number_count ▼	stock_cnt ▼
1	5	3676
2	0	7
3	1	1

```

SELECT DISTINCT StockCode, number_count
FROM (
    SELECT StockCode,
        LENGTH(StockCode) - LENGTH(REGEXP_REPLACE(StockCode, r'[0-9]', '')) AS
number_count
    FROM `modulabs_project.data`
)
WHERE number_count <= 1;

```

행	StockCode	number_count
1	POST	0
2	M	0
3	C2	1
4	D	0
5	BANK CHARGES	0
6	PADS	0
7	DOT	0
8	CRUK	0

```

SELECT
    ROUND(SUM(CASE WHEN StockCode IN ('POST', 'D', 'C2', 'M', 'BANK CHARGES',
'PADS', 'DOT', 'CRUK')
        THEN 1 ELSE 0 END) / COUNT(*) * 100, 2) AS outlier_percentage
FROM `modulabs_project.data`;

```

행	outlier_percentage
1	0.48


```

DELETE FROM `modulabs_project.data`
WHERE StockCode IN (
  SELECT DISTINCT StockCode
  FROM (
    SELECT StockCode,
           LENGTH(StockCode) - LENGTH(REGEXP_REPLACE(StockCode, r'[0-9]', ''))
    AS number_count
    FROM `modulabs_project.data`
  )
  WHERE number_count <= 1
);

```



이 문으로 data의 행 1,915개가 삭제되었습니다.

```

SELECT Description, COUNT(*) AS description_cnt
FROM `modulabs_project.data`
GROUP BY Description
ORDER BY description_cnt DESC
LIMIT 30;

```

행	Description	description_cnt
1	WHITE HANGING HEART T-LIG...	2058
2	REGENCY CAKESTAND 3 TIER	1894
3	JUMBO BAG RED RETROSPOT	1659
4	PARTY BUNTING	1409
5	ASSORTED COLOUR BIRD ORN...	1405
6	LUNCH BAG RED RETROSPOT	1345
7	SET OF 3 CAKE TINS PANTRY D...	1224
8	LUNCH BAG BLACK SKULL	1099
9	PACK OF 72 RETROSPOT CAKE ...	1062
10	SPOTTY BUNTING	1026
11	PAPER CHAIN KIT 50'S CHRIST...	1013
12	LUNCH BAG SPACEBOY DESIGN	1006
13	LUNCH BAG CARS BLUE	1000
14	HEART OF WICKER SMALL	990
15	NATURAL SLATE HEART CHAL...	989
16	JAM MAKING SET WITH JARS	966

```
SELECT DISTINCT Description
FROM `modulabs_project.data`
WHERE REGEXP_CONTAINS(Description, r'[a-z]');
```

행	Description ▼
1	BAG 250g SWIRLY MARBLES
2	3 TRADITIONAL BISCUIT CUTTE...
3	BAG 125g SWIRLY MARBLES
4	POLYESTER FILLER PAD 30CMx...
5	BAG 500g SWIRLY MARBLES
6	POLYESTER FILLER PAD 45x45...
7	POLYESTER FILLER PAD 40x40...
8	ESSENTIAL BALM 3.5g TIN IN E...
9	High Resolution Image
10	FRENCH BLUE METAL DOOR SI...
11	POLYESTER FILLER PAD 45x30...
12	Next Day Carriage
13	NUMBER TILE VINTAGE FONT No
14	NUMBER TILE COTTAGE GARD...
15	POLYESTER FILLER PAD 65CMx...
16	FLOWERS HANDBAG blue and o...

페이지당 결과 수: 50 ▼ 1 - 19 (전체 19행) |< < > >|

```
DELETE
FROM `modulabs_project.data`
WHERE Description IN (
  'Next Day Carriage',
  'High Resolution Image',
  'Manual',
  'DOTCOM POSTAGE',
  'Amazon Adjustment',
  'Adjust bad debt',
  'CARRIAGE'
);
```

i 이 문으로 data의 행 83개가 삭제되었습니다.

```
CREATE OR REPLACE TABLE `modulabs_project.data` AS
SELECT
  * EXCEPT (Description),
  UPPER(Description) AS Description
FROM `modulabs_project.data`;
```

i 이 문으로 이름이 data인 테이블이 교체되었습니다.

```

SELECT
    MIN(UnitPrice) AS min_price,
    MAX(UnitPrice) AS max_price,
    ROUND(AVG(UnitPrice), 2) AS avg_price
FROM `modulabs_project.data`;

```

행	min_price	max_price	avg_price
1	0.0	649.5	2.9

```

SELECT
    COUNT(*) AS cnt_quantity,
    MIN(Quantity) AS min_quantity,
    MAX(Quantity) AS max_quantity,
    ROUND(AVG(Quantity), 2) AS avg_quantity
FROM `modulabs_project.data`
WHERE UnitPrice = 0;

```

행	cnt_quantity	min_quantity	max_quantity	avg_quantity
1	33	1	12540	420.52

```

CREATE OR REPLACE TABLE `modulabs_project.data` AS
SELECT *
FROM `modulabs_project.data`
WHERE UnitPrice > 0;

```



이 문으로 이름이 data인 테이블이 교체되었습니다.

```
SELECT DATE(InvoiceDate) AS InvoiceDay, *
FROM `modulabs_project.data`;
```

행	InvoiceDay	InvoiceNo	StockCode	Quantity	InvoiceDate	UnitPrice	CustomerID
1	2011-01-18	541431	23166	74215	2011-01-18 10:01:00 UTC	1.04	
2	2011-01-18	C541433	23166	-74215	2011-01-18 10:17:00 UTC	1.04	
3	2010-12-07	537626	84997C	6	2010-12-07 14:57:00 UTC	3.75	
4	2010-12-07	537626	22497	4	2010-12-07 14:57:00 UTC	4.25	
5	2010-12-07	537626	22375	4	2010-12-07 14:57:00 UTC	4.25	
6	2010-12-07	537626	22771	12	2010-12-07 14:57:00 UTC	1.25	
7	2010-12-07	537626	22195	12	2010-12-07 14:57:00 UTC	1.65	
8	2010-12-07	537626	22492	36	2010-12-07 14:57:00 UTC	0.65	
9	2010-12-07	537626	22494	12	2010-12-07 14:57:00 UTC	1.25	
10	2010-12-07	537626	84969	6	2010-12-07 14:57:00 UTC	4.25	
11	2010-12-07	537626	22772	12	2010-12-07 14:57:00 UTC	1.25	
12	2010-12-07	537626	85167B	30	2010-12-07 14:57:00 UTC	1.25	
13	2010-12-07	537626	21064	6	2010-12-07 14:57:00 UTC	5.95	
14	2010-12-07	537626	22212	6	2010-12-07 14:57:00 UTC	2.1	
15	2010-12-07	537626	21171	12	2010-12-07 14:57:00 UTC	1.45	
16	2010-12-07	537626	21731	12	2010-12-07 14:57:00 UTC	1.65	

페이지당 결과 수: 50 1 - 50 (전체 399573행) |< < > >|

```
SELECT
    MAX(DATE(InvoiceDate)) OVER() AS most_recent_date,
    DATE(InvoiceDate) AS InvoiceDay,
    *
FROM `modulabs_project.data`;
```

행	most_recent_date	InvoiceDay	InvoiceNo	StockCode	Quantity	InvoiceDate	UnitPrice
1	2011-12-09	2011-10-18	571705	23008	2	2011-10-18 15:22:00 UTC	
2	2011-12-09	2011-09-28	568651	22947	2	2011-09-28 12:04:00 UTC	
3	2011-12-09	2011-08-02	562046	47590B	3	2011-08-02 10:34:00 UTC	
4	2011-12-09	2011-03-03	545475	21984	432	2011-03-03 10:59:00 UTC	
5	2011-12-09	2011-08-18	563614	22749	48	2011-08-18 08:51:00 UTC	
6	2011-12-09	2011-11-30	579692	22693	192	2011-11-30 14:05:00 UTC	
7	2011-12-09	2011-08-22	563950	23375	10	2011-08-22 10:39:00 UTC	
8	2011-12-09	2011-11-07	574885	23084	48	2011-11-07 14:01:00 UTC	
9	2011-12-09	2011-10-17	571312	84251G	12	2011-10-17 10:42:00 UTC	
10	2011-12-09	2011-08-31	564856	35599B	12	2011-08-31 09:11:00 UTC	
11	2011-12-09	2011-08-31	564856	23094	3	2011-08-31 09:11:00 UTC	
12	2011-12-09	2011-05-27	554940	23108	2	2011-05-27 13:49:00 UTC	
13	2011-12-09	2011-11-11	575886	22059	8	2011-11-11 13:57:00 UTC	
14	2011-12-09	2011-03-08	545916	21843	5	2011-03-08 11:46:00 UTC	
15	2011-12-09	2011-08-01	561911	23108	2	2011-08-01 10:26:00 UTC	
16	2011-12-09	2011-01-11	540785	84997D	360	2011-01-11 11:39:00 UTC	

페이지당 결과 수: 50 1 - 50 (전체 399573행) |< < > >|

```

SELECT
    CustomerID,
    MAX(Date(InvoiceDate)) AS InvoiceDay
FROM `modulabs_project.data`
GROUP BY CustomerID;

```

행	CustomerID	InvoiceDay
1	12346	2011-01-18
2	12347	2011-12-07
3	12348	2011-09-25
4	12349	2011-11-21
5	12350	2011-02-02
6	12352	2011-11-03
7	12353	2011-05-19
8	12354	2011-04-21
9	12355	2011-05-09
10	12356	2011-11-17
11	12357	2011-11-06
12	12358	2011-12-08
13	12359	2011-12-02
14	12360	2011-10-18
15	12361	2011-02-25
16	12362	2011-12-06

페이지당 결과 수: 50 1 - 50 (전체 4362행)

```

SELECT
    CustomerID,
    EXTRACT(DAY FROM MAX(InvoiceDay) OVER () - InvoiceDay) AS recency
FROM (
    SELECT
        CustomerID,
        MAX(Date(InvoiceDate)) AS InvoiceDay
    FROM ecstatic-valve-465201-c2.modulabs_project.data
    GROUP BY CustomerID
);

```

행	CustomerID	recency
1	12363	109
2	12397	35
3	12520	79
4	12614	277
5	12851	96
6	12854	78
7	12875	143
8	12952	5
9	13015	53
10	13118	19
11	13235	212
12	13670	75
13	13807	367
14	13837	211
15	13951	176
16	14064	29

페이지당 결과 수: 50 1 - 50 (전체 4362행)

```

CREATE OR REPLACE TABLE `modulabs_project.user_r` AS
SELECT
  CustomerID,
  EXTRACT(DAY FROM MAX(InvoiceDay) OVER () - InvoiceDay) AS recency
FROM (
  SELECT
    CustomerID,
    MAX(DATE(InvoiceDate)) AS InvoiceDay
  FROM `modulabs_project.data`
  GROUP BY CustomerID
);

```



이 문으로 이름이 user_r인 테이블이 교체되었습니다.

```

CREATE OR REPLACE TABLE `ecstatic-valve-465201-c2.modulabs_project.user_r` AS
SELECT
  CustomerID,
  EXTRACT(DAY FROM MAX(InvoiceDay) OVER () - InvoiceDay) AS recency
FROM (
  SELECT
    CustomerID,
    MAX(DATE(InvoiceDate)) AS InvoiceDay
  FROM `ecstatic-valve-465201-c2.modulabs_project.data`
  GROUP BY CustomerID
);

```



이 문으로 이름이 user_r인 테이블이 교체되었습니다.

```
SELECT
    CustomerID,
    COUNT(DISTINCT InvoiceNo) AS purchase_cnt
FROM `ecstatic-valve-465201-c2.modulabs_project.data`
GROUP BY CustomerID;
```

행	CustomerID	purchase_cnt
1	12346	2
2	12347	7
3	12348	4
4	12349	1
5	12350	1
6	12352	8
7	12353	1
8	12354	1
9	12355	1
10	12356	3
11	12357	1
12	12358	2
13	12359	6
14	12360	3
15	12361	1
16	12362	13

페이지당 결과 수: 50 1 - 50 (전체 4362행) |< < > >|

```
SELECT
    CustomerID,
    SUM(Quantity) AS item_cnt
FROM `ecstatic-valve-465201-c2.modulabs_project.data`
GROUP BY CustomerID;
```

행	CustomerID	item_cnt
1	12346	0
2	12347	2458
3	12348	2332
4	12349	630
5	12350	196
6	12352	463
7	12353	20
8	12354	530
9	12355	240
10	12356	1573
11	12357	2708
12	12358	242
13	12359	1599
14	12360	1156
15	12361	90
16	12362	2180

페이지당 결과 수: 50 1 - 50 (전체 4362행) |< < > >|

```
CREATE OR REPLACE TABLE `ecstatic-valve-465201-c2.modulabs_project.user_rf`  
AS
```

```
-- (1) 전체 거래 건수 계산
```

```
WITH purchase_cnt AS (  
    SELECT  
        CustomerID,  
        COUNT(DISTINCT InvoiceNo) AS purchase_cnt  
    FROM `ecstatic-valve-465201-c2.modulabs_project.data`  
    GROUP BY CustomerID  
)
```

```
-- (2) 구매한 아이템 총 수량 계산
```

```
item_cnt AS (  
    SELECT  
        CustomerID,  
        SUM(Quantity) AS item_cnt  
    FROM `ecstatic-valve-465201-c2.modulabs_project.data`  
    GROUP BY CustomerID  
)
```

```
-- 기존 user_r 테이블과 결합
```

```
SELECT  
    pc.CustomerID,  
    pc.purchase_cnt,  
    ic.item_cnt,  
    ur.recency  
FROM purchase_cnt AS pc  
JOIN item_cnt AS ic  
    ON pc.CustomerID = ic.CustomerID  
JOIN `ecstatic-valve-465201-c2.modulabs_project.user_r` AS ur  
    ON pc.CustomerID = ur.CustomerID;
```



이 문으로 이름이 user_rf인 테이블이 교체되었습니다.


```

SELECT
    CustomerID,
    ROUND(SUM(UnitPrice * Quantity), 1) AS user_total
FROM `ecstatic-valve-465201-c2.modulabs_project.data`
GROUP BY CustomerID;

```

행	CustomerID	user_total
1	12346	0.0
2	12347	4310.0
3	12348	1437.2
4	12349	1457.6
5	12350	294.4
6	12352	1265.4
7	12353	89.0
8	12354	1079.4
9	12355	459.4
10	12356	2487.4
11	12357	6207.7
12	12358	928.1
13	12359	6183.0
14	12360	2302.1
15	12361	174.9
16	12362	4665.6

페이지당 결과 수: 50 1 ~ 50 (전체 4362행) |< < > >|

```

CREATE OR REPLACE TABLE `ecstatic-valve-465201-c2.modulabs_project.user_rfm`
AS
SELECT
    rf.CustomerID AS CustomerID,
    rf.purchase_cnt,
    rf.item_cnt,
    rf.recency,
    ut.user_total,
    ROUND(ut.user_total / rf.purchase_cnt, 1) AS user_average
FROM `ecstatic-valve-465201-c2.modulabs_project.user_rf` rf
LEFT JOIN (
    -- 고객 별 총 지출액
    SELECT
        CustomerID,
        ROUND(SUM(UnitPrice * Quantity), 1) AS user_total
    FROM `ecstatic-valve-465201-c2.modulabs_project.data`
    GROUP BY CustomerID
) ut
ON rf.CustomerID = ut.CustomerID;

```



이 문으로 이름이 user_rfm인 테이블이 교체되었습니다.

```
SELECT COUNT(DISTINCT CustomerID) AS unique_users
FROM `ecstatic-valve-465201-c2.modulabs_project.user_rfm`;
```

행	unique_users
1	4362

```
CREATE OR REPLACE TABLE `ecstatic-valve-465201-c2.modulabs_project.user_data`
AS
WITH unique_products AS (
  SELECT
    CustomerID,
    COUNT(DISTINCT StockCode) AS unique_products
  FROM `ecstatic-valve-465201-c2.modulabs_project.data`
  GROUP BY CustomerID
)
```

```
SELECT
  ur.*,
  up.unique_products
FROM `ecstatic-valve-465201-c2.modulabs_project.user_rfm` AS ur
JOIN unique_products AS up
  ON ur.CustomerID = up.CustomerID;
```

i 이 문으로 이름이 user_data인 테이블이 교체되었습니다.

```
CREATE OR REPLACE TABLE `ecstatic-valve-465201-c2.modulabs_project.user_data`
AS
WITH purchase_intervals AS (
  -- (2) 고객 별 구매와 구매 사이의 평균 소요 일수
  SELECT
    CustomerID,
    CASE
      WHEN ROUND(AVG(interval_), 2) IS NULL THEN 0
      ELSE ROUND(AVG(interval_), 2)
    END AS average_interval
  FROM (
```

```

-- (1) 구매와 구매 사이에 소요된 일수
SELECT
    CustomerID,
    DATE_DIFF(DATE(InvoiceDate),
              LAG(DATE(InvoiceDate)) OVER (PARTITION BY CustomerID ORDER
BY DATE(InvoiceDate)),
              DAY) AS interval_
FROM
    `ecstatic-valve-465201-c2.modulabs_project.data`
WHERE CustomerID IS NOT NULL
)
GROUP BY CustomerID
)

```

```

SELECT
    u.*,
    pi.average_interval
FROM `ecstatic-valve-465201-c2.modulabs_project.user_data` AS u
LEFT JOIN purchase_intervals AS pi
ON u.CustomerID = pi.CustomerID;

```



이 문으로 이름이 user_data인 테이블이 교체되었습니다.

```

CREATE OR REPLACE TABLE ecstatic-valve-465201-c2.modulabs_project.user_data
AS

```

```

WITH TransactionInfo AS (
    SELECT
        CustomerID,
        COUNT(DISTINCT InvoiceNo) AS total_transactions,
        COUNT(DISTINCT CASE WHEN InvoiceNo LIKE 'C%' THEN InvoiceNo END) AS
cancel_frequency
    FROM ecstatic-valve-465201-c2.modulabs_project.data
    WHERE CustomerID IS NOT NULL
    GROUP BY CustomerID
)

```

```

SELECT
    u.*,
    t.cancel_frequency,

```

```

ROUND(t.cancel_frequency / t.total_transactions, 2) AS cancel_rate
FROM ecstatic-valve-465201-c2.modulabs_project.user_data AS u
LEFT JOIN TransactionInfo AS t
ON u.CustomerID = t.CustomerID;

```

i 이 문으로 이름이 user_data인 테이블이 교체되었습니다.

```

SELECT *
FROM `ecstatic-valve-465201-c2.modulabs_project.user_data`;

```

행	CustomerID	purchase_cnt	item_cnt	recency	user_total	user_average	unique_products	average_interval	cancel_fr
1	15992	1	17	3	42.0	42.0	3	0.0	
2	13194	1	26	135	60.7	60.7	3	0.0	
3	14816	1	5	197	271.8	271.8	4	0.0	
4	12968	1	73	112	135.9	135.9	5	0.0	
5	13095	1	144	70	74.4	74.4	6	0.0	
6	18191	1	140	261	207.8	207.8	7	0.0	
7	13484	1	44	241	313.8	313.8	8	0.0	
8	12551	1	100	357	168.0	168.0	10	0.0	
9	15895	1	94	149	179.2	179.2	11	0.0	
10	13898	1	133	325	155.9	155.9	12	0.0	
11	14523	1	251	108	241.1	241.1	12	0.0	
12	16756	1	106	214	239.4	239.4	13	0.0	
13	13127	1	127	77	259.2	259.2	15	0.0	
14	13889	1	146	92	416.9	416.9	15	0.0	
15	16183	1	113	67	338.7	338.7	15	0.0	
16	12561	1	177	302	238.8	238.8	16	0.0	

페이지당 결과 수: 50 1 ~ 50 (전체 4362행) |< < > >|