データ分析のためのSQL勉強会
～実践編 ハンズオン～

Agenda

アジェンダ

- SQL勉強会 〜実践編〜 概要説明
  1. 対象者
  2. ゴール
  3. 進め方
- SQL問題集の実践 + 解説

Agenda

アジェンダ

- SQL勉強会 〜実践編〜 概要説明
  1. 対象者
  2. ゴール
  3. 進め方

- SQL問題集の実践 + 解説

1. 対象者

- SQLの基礎を理解している人
  - 『SQL勉強会～初級編～』を受講している人
  - 『データ分析力を高めるビジネスパーソンのためのSQL入門書』を読んだ人

- 実務でデータ分析を使えるようになりたい人

※今回は「データ分析のための」SQL勉強会です
※アプリケーション開発などで使うSQLは想定していないのでデータ取得(SELECT文)が中心になっています

2. ゴール

- データ分析の実務で使うSQLを理解する
- データ分析の実務で使うSQLを自分で書けるようにする

※ゼロベースでSQLをかける様になる必要はない

- 調べてかける様にする
- 他の人が書いたSQL(自分が過去に書いたSQL)を参考にSQLの読み解きができるようにする

3. 進め方

- 初級編と同様にハンズオン形式で実施
  - ※初級編で用意した環境でSQLの実行環境が整っていること
- 実際のデータ分析で使う集計をもとに計10問の実践問題を実施
- 実践問題をベースに基礎知識を説明後自分で考えてSQLを書いていくスタイル
- 問題の解答例も一緒に載せていますが、解答をみる前にまずは自分で考えてみましょう

Agenda

アジェンダ

- SQL勉強会 ～実践編～ 概要説明
  1. 対象者
  2. ゴール
  3. 進め方

- **SQL問題集の実践 + 解説**

実践SQL問題集　10選

No	問題	難易度
1	性別ごとの人数を集計してください	★☆☆☆☆
2	年代別の人数を集計してください	★★★☆☆
3	日別の売上と購入件数を集計してください	★★☆☆☆
4	月別の売上と購入件数を集計してください	★★☆☆☆
5	ユーザーごとの最終購入日/購入頻度/購入金額を集計してください	★★★☆☆
6	購入頻度(注文日数)ごとにユーザー数を集計してください①	★★★★☆
7	購入頻度(注文日数)ごとにユーザー数を集計してください②	★★★★★
8	「食品」を購入したユーザーIDをユニーク(重複なし)で抽出してください	★★☆☆☆
9	「美容」を買っている人が他に何を買っているか抽出して下さい	★★★★☆
10	初回購入で何が一番購入されているか抽出して下さい	★★★★★

サンプルデータ

usersテーブル

カラム名	意味	型	備考
user_id	ユーザーID	TEXT	
gender	性別	TEXT	女性 / 男性
birth	誕生日	INTEGER	誕生年が数字で入ります
is_deleted	削除フラグ	INTEGER	0：通常ユーザー 1：削除ユーザー

productsテーブル

カラム名	意味	型	備考
product_id	商品ID	TEXT	
name	商品名	TEXT	
price	金額	INTEGER	
large_category	大カテゴリ	TEXT	
medium_category	中カテゴリ	TEXT	
small_category	小カテゴリ	TEXT	

今回は架空のECサイトの購買データを参考にデータ抽出を行います

サンプルデータ：
https://github.com/hikarut/SQL-Sample-Data

ordersテーブル

カラム名	意味	型	備考
order_id	注文ID	TEXT	
user_id	ユーザーID	TEXT	
order_product_id	商品ID	TEXT	
order_date	注文日時	INTEGER	
is_discounted	割引フラグ	INTEGER	0：割引なし 1：割引あり
is_canceled	キャンセルフラグ	INTEGER	0：キャンセルなし 1：キャンセルあり

実践SQL問題集　10選

No	問題	難易度
1	性別ごとの人数を集計してください	★☆☆☆☆
2	年代別の人数を集計してください	★★★☆☆
3	日別の売上と購入件数を集計してください	★★☆☆☆
4	月別の売上と購入件数を集計してください	★★☆☆☆
5	ユーザーごとの最終購入日/購入頻度/購入金額を集計してください	★★★☆☆
6	購入頻度(注文日数)ごとにユーザー数を集計してください①	★★★★☆
7	購入頻度(注文日数)ごとにユーザー数を集計してください②	★★★★★
8	「食品」を購入したユーザーIDをユニーク(重複なし)で抽出してください	★★☆☆☆
9	「美容」を買っている人が他に何を買っているか抽出して下さい	★★★★☆
10	初回購入で何が一番購入されているか抽出して下さい	★★★★★

実践SQL問題①

難易度	★☆☆☆☆
所要時間	3分

問題

性別ごとの人数を集計してください。ただし削除済ユーザーは除外してください。

実践SQL問題①

難易度	★☆☆☆☆
所要時間	3分

問題

性別ごとの人数を集計してください。ただし削除済ユーザーは除外してください。

ヒント①	usersテーブルに性別が分かるカラムが入ってます

実践SQL問題①

難易度	★☆☆☆☆
所要時間	3分

問題

性別ごとの人数を集計してください。ただし削除済ユーザーは除外してください。

ヒント①	usersテーブルに性別が分かるカラムが入ってます
ヒント②	特定の属性ごとにカウントしたい場合はGROUP BYを使いましょう

実践SQL問題①

難易度	★☆☆☆☆
所要時間	3分

問題

性別ごとの人数を集計してください。ただし削除済ユーザーは除外してください。

ヒント①	usersテーブルに性別が分かるカラムが入ってます
ヒント②	特定の属性ごとにカウントしたい場合はGROUP BYを使いましょう
ヒント③	『削除済ユーザー』はusersテーブルのis_deletedフラグで判定できます

実践SQL問題①　解答例

難易度	★☆☆☆☆
所要時間	3分

問題

性別ごとの人数を集計してください。ただし削除済ユーザーは除外してください。

```sql
SELECT
    gender,
    COUNT(*)
FROM users
WHERE is_deleted = 0
GROUP BY gender
```

実践SQL問題①　解答例

難易度	★☆☆☆☆
所要時間	3分

問題

性別ごとの人数を集計してください。ただし削除済ユーザーは除外してください。

```sql
SELECT
    gender,
    COUNT(*)
FROM users
WHERE is_deleted = 0
GROUP BY gender
```

- group byを使って性別ごとにグルーピング
- COUNT(*)で人数を集計
  COUNT(DISTINCT user_id)でもOK
- WHEREの条件で削除済ユーザーを除外

実践SQL問題集　10選

No	問題	難易度
1	性別ごとの人数を集計してください	★☆☆☆☆
2	年代別の人数を集計してください	★★★☆☆
3	日別の売上と購入件数を集計してください	★★☆☆☆
4	月別の売上と購入件数を集計してください	★★☆☆☆
5	ユーザーごとの最終購入日/購入頻度/購入金額を集計してください	★★★☆☆
6	購入頻度(注文日数)ごとにユーザー数を集計してください①	★★★★☆
7	購入頻度(注文日数)ごとにユーザー数を集計してください②	★★★★★
8	「食品」を購入したユーザーIDをユニーク(重複なし)で抽出してください	★★☆☆☆
9	「美容」を買っている人が他に何を買っているか抽出して下さい	★★★★☆
10	初回購入で何が一番購入されているか抽出して下さい	★★★★★

実践SQL問題②

難易度	★★★☆☆
所要時間	5分

問題

以下のSQLを参考に年代別の人数を集計してください。削除済ユーザーは除外して、年代は10代、20代、30代、、、80代と10区切りで計算してください。

```sql
SELECT
    user_id,
    gender,
    birth,
    date(),
    substr(date(), 1, 4) as year,
    substr(date(), 1, 4) - birth
FROM users
```

関数説明

date()	現在日付を取得する関数
substr()	文字列を切り取る関数 使い方：substr(切り取る文字列, 開始位置, 文字数)
substr(date(), 1, 4)	現在日付の文字列から1文字目から4文字切り抜く 例：2022-06-27→2022
substr(date(), 1, 4) - birth	現在の西暦から誕生年の西暦を引き算する→今年の年齢を取得

実践SQL問題②

難易度	★★★☆☆
所要時間	5分

問題

以下のSQLを参考に年代別の人数を集計してください。削除済ユーザーは除外して、年代は10代、20代、30代、、、80代と10区切りで計算してください。

ヒント①	まずはcase文を使って「年代」のカラムを作ってみましょう

実践SQL問題②

難易度	★★★☆☆
所要時間	5分

問題

以下のSQLを参考に年代別の人数を集計してください。削除済ユーザーは除外して、年代は10代、20代、30代、、、80代と10区切りで計算してください。

ヒント①	まずはcase文を使って「年代」のカラムを作ってみましょう
ヒント②	WITH句(サブクエリ)を使って「年代」を追加した一時テーブルを作成してみましょう

実践SQL問題②

難易度	★★★☆☆
所要時間	5分

問題

以下のSQLを参考に年代別の人数を集計してください。削除済ユーザーは除外して、年代は10代、20代、30代、、、80代と10区切りで計算してください。

ヒント①	まずはcase文を使って「年代」のカラムを作ってみましょう
ヒント②	WITH句(サブクエリ)を使って「年代」を追加した一時テーブルを作成してみましょう
ヒント③	「年代」でGROUP BYしてカウントしてみましょう

実践SQL問題②　解答例

難易度	★★★☆☆
所要時間	5分

```sql
WITH user_add_age AS(
    SELECT
        user_id,
        gender,
        birth,
        substr(date(), 1, 4) - birth AS age,
        CASE
            WHEN substr(date(), 1, 4) - birth >= 80 THEN '80代'
            WHEN substr(date(), 1, 4) - birth >= 70 THEN '70代'
            WHEN substr(date(), 1, 4) - birth >= 60 THEN '60代'
            WHEN substr(date(), 1, 4) - birth >= 50 THEN '50代'
            WHEN substr(date(), 1, 4) - birth >= 40 THEN '40代'
            WHEN substr(date(), 1, 4) - birth >= 30 THEN '30代'
            WHEN substr(date(), 1, 4) - birth >= 20 THEN '20代'
            WHEN substr(date(), 1, 4) - birth < 20 THEN '10代'
            ELSE 'その他'
        END AS age_range
    FROM users
    WHERE is_deleted = 0
)

SELECT
    age_range,
    COUNT(DISTINCT user_id) as uu
FROM user_add_age
GROUP BY age_range
ORDER BY age_range
```

実践SQL問題②　解答例

難易度	★★★☆☆
所要時間	5分

```sql
WITH user_add_age AS(
    SELECT
        user_id,
        gender,
        birth,
        substr(date(), 1, 4) - birth AS age,
        CASE
            WHEN substr(date(), 1, 4) - birth >= 80 THEN '80代'
            WHEN substr(date(), 1, 4) - birth >= 70 THEN '70代'
            WHEN substr(date(), 1, 4) - birth >= 60 THEN '60代'
            WHEN substr(date(), 1, 4) - birth >= 50 THEN '50代'
            WHEN substr(date(), 1, 4) - birth >= 40 THEN '40代'
            WHEN substr(date(), 1, 4) - birth >= 30 THEN '30代'
            WHEN substr(date(), 1, 4) - birth >= 20 THEN '20代'
            WHEN substr(date(), 1, 4) - birth < 20 THEN '10代'
            ELSE 'その他'
        END AS age_range
    FROM users
    WHERE is_deleted = 0
)

SELECT
    age_range,
    COUNT(DISTINCT user_id) as uu
FROM user_add_age
GROUP BY age_range
ORDER BY age_range
```

- 年齢は「substr(date(), 1, 4) - birth」で計算
  - ※「age」は今回は直接使わないのでカラムとしてなくても問題なし
- 年齢をCASE式を使って年代に変換

実践SQL問題②　解答例

難易度	★★★☆☆
所要時間	5分

```sql
WITH user_add_age AS(
    SELECT
        user_id,
        gender,
        birth,
        substr(date(), 1, 4) - birth AS age,
        CASE
            WHEN substr(date(), 1, 4) - birth >= 80 THEN '80代'
            WHEN substr(date(), 1, 4) - birth >= 70 THEN '70代'
            WHEN substr(date(), 1, 4) - birth >= 60 THEN '60代'
            WHEN substr(date(), 1, 4) - birth >= 50 THEN '50代'
            WHEN substr(date(), 1, 4) - birth >= 40 THEN '40代'
            WHEN substr(date(), 1, 4) - birth >= 30 THEN '30代'
            WHEN substr(date(), 1, 4) - birth >= 20 THEN '20代'
            WHEN substr(date(), 1, 4) - birth < 20 THEN '10代'
            ELSE 'その他'
        END AS age_range
    FROM users
    WHERE is_deleted = 0
)

SELECT
    age_range,
    COUNT(DISTINCT user_id) as uu
FROM user_add_age
GROUP BY age_range
ORDER BY age_range
```

- 年代を入れた情報で一時テーブルを作成
- 削除済ユーザーは除外する

実践SQL問題②　解答例

難易度　★★★☆☆

所要時間　5分

```sql
WITH user_add_age AS(
    SELECT
        user_id,
        gender,
        birth,
        substr(date(), 1, 4) - birth AS age,
        CASE
            WHEN substr(date(), 1, 4) - birth >= 80 THEN '80代'
            WHEN substr(date(), 1, 4) - birth >= 70 THEN '70代'
            WHEN substr(date(), 1, 4) - birth >= 60 THEN '60代'
            WHEN substr(date(), 1, 4) - birth >= 50 THEN '50代'
            WHEN substr(date(), 1, 4) - birth >= 40 THEN '40代'
            WHEN substr(date(), 1, 4) - birth >= 30 THEN '30代'
            WHEN substr(date(), 1, 4) - birth >= 20 THEN '20代'
            WHEN substr(date(), 1, 4) - birth < 20 THEN '10代'
            ELSE 'その他'
        END AS age_range
    FROM users
    WHERE is_deleted = 0
)

SELECT
    age_range,
    COUNT(DISTINCT user_id) as uu
FROM user_add_age
GROUP BY age_range
ORDER BY age_range
```

- 年代(age_range)でGROUP BYして人数をカウント
- 人数のカウントはuser_idをユニークにしてカウントする
- 必要に応じてORDER BYで年代順に並べ替え

実践SQL問題②　解説

元のテーブルにない情報で集計したい場合はWITH句を使って一時テーブルを作成することで順番にデータを整理して最終的に集計したいデータをまとめることができる

STEP 1

現状のテーブルの確認

カラム名	意味
user_id	ユーザーID
gender	性別
birth	誕生日
is_deleted	削除フラグ

STEP 2

集計ように新しいカラム（年代）を追加

カラム名	意味
user_id	ユーザーID
gender	性別
birth	誕生日
is_deleted	削除フラグ
age_range	年代

STEP 3

追加したカラムでグルーピングして集計

カラム名	意味
age_range	年代
COUNT(user_id)	ユーザーIDの数

実践SQL問題集　10選

No	問題	難易度
1	性別ごとの人数を集計してください	★☆☆☆☆
2	年代別の人数を集計してください	★★★☆☆
3	日別の売上と購入件数を集計してください	★★☆☆☆
4	月別の売上と購入件数を集計してください	★★☆☆☆
5	ユーザーごとの最終購入日/購入頻度/購入金額を集計してください	★★★☆☆
6	購入頻度(注文日数)ごとにユーザー数を集計してください①	★★★★☆
7	購入頻度(注文日数)ごとにユーザー数を集計してください②	★★★★★
8	「食品」を購入したユーザーIDをユニーク(重複なし)で抽出してください	★★☆☆☆
9	「美容」を買っている人が他に何を買っているか抽出して下さい	★★★★☆
10	初回購入で何が一番購入されているか抽出して下さい	★★★★★

実践SQL問題③

難易度	★★☆☆☆
所要時間	5分

問題

日別の売上と購入件数を集計してください。ただし注文キャンセルのデータは除外してください。

実践SQL問題③

難易度	★★☆☆☆
所要時間	5分

問題

日別の売上と購入件数を集計してください。ただし注文キャンセルのデータは除外してください。

ヒント①

売上を集計するためには注文情報と商品情報を結合(JOIN)させる必要があります

実践SQL問題③

難易度	★★☆☆☆
所要時間	5分

問題

日別の売上と購入件数を集計してください。ただし注文キャンセルのデータは除外してください。

ヒント①	売上を集計するためには注文情報と商品情報を結合(JOIN)させる必要があります
ヒント②	注文情報(orders)と商品情報(products)は商品ID(product_id)を使って結合できます

実践SQL問題③

難易度	★★☆☆☆
所要時間	5分

問題

日別の売上と購入件数を集計してください。ただし注文キャンセルのデータは除外してください。

ヒント①	売上を集計するためには注文情報と商品情報を結合(JOIN)させる必要があります
ヒント②	注文情報(orders)と商品情報(products)は商品ID(product_id)を使って結合できます
ヒント③	日付でGROUP BYして売上と購入件数を集計しましょう

実践SQL問題③　解答例

難易度	★★☆☆☆
所要時間	5分

```sql
SELECT
    o.order_date,
    COUNT(*),
    SUM(p.price)
FROM orders AS o
LEFT JOIN products AS p ON o.order_product_id = p.product_id
WHERE o.is_canceled = 0
GROUP BY o.order_date
ORDER BY o.order_date
```

実践SQL問題③　解答例

難易度	★★☆☆☆
所要時間	5分

```sql
SELECT
    o.order_date,
    COUNT(*),
    SUM(p.price)
FROM orders AS o
LEFT JOIN products AS p ON o.order_product_id = p.product_id
WHERE o.is_canceled = 0
GROUP BY o.order_date
ORDER BY o.order_date
```

- productsのproduct_idとordersのorder_product_idを使って商品情報と注文情報を結合させる
- 結合の際はINNER JOINかLEFT JOINを使う
  - ※今回はINNER JOINでもLEFT JOINでも結果は同じ

実践SQL問題③　解答例

難易度	★★☆☆☆
所要時間	5分

```sql
SELECT
    o.order_date,
    COUNT(*),
    SUM(p.price)
FROM orders AS o
LEFT JOIN products AS p ON o.order_product_id = p.product_id
WHERE o.is_canceled = 0
GROUP BY o.order_date
ORDER BY o.order_date
```

- 日付情報でGROUP BYをして売上と購入件数を集計
- 売上はpriceの合計、購入件数はレコード数(注文情報の行数)をカウントする
- is_canceled = 0で注文キャンセルのデータを除外

実践SQL問題集　10選

No	問題	難易度
1	性別ごとの人数を集計してください	★☆☆☆☆
2	年代別の人数を集計してください	★★★☆☆
3	日別の売上と購入件数を集計してください	★★☆☆☆
4	月別の売上と購入件数を集計してください	★★☆☆☆
5	ユーザーごとの最終購入日/購入頻度/購入金額を集計してください	★★★☆☆
6	購入頻度(注文日数)ごとにユーザー数を集計してください①	★★★★☆
7	購入頻度(注文日数)ごとにユーザー数を集計してください②	★★★★★
8	「食品」を購入したユーザーIDをユニーク(重複なし)で抽出してください	★★☆☆☆
9	「美容」を買っている人が他に何を買っているか抽出して下さい	★★★★☆
10	初回購入で何が一番購入されているか抽出して下さい	★★★★★

実践SQL問題④

難易度	★★☆☆☆
所要時間	5分

問題

月別の売上と購入件数を集計してください。ただし注文キャンセルのデータは除外してください。

実践SQL問題④

難易度	★★☆☆☆
所要時間	5分

問題

月別の売上と購入件数を集計してください。ただし注文キャンセルのデータは除外してください。

ヒント①

「月」情報の取得はsubstrを使って日付から「月」に変換しましょう

実践SQL問題④

難易度	★★☆☆☆
所要時間	5分

問題

月別の売上と購入件数を集計してください。ただし注文キャンセルのデータは除外してください。

ヒント①	「月」情報の取得はsubstrを使って日付から「月」に変換しましょう
ヒント②	「月」情報は「substr(order_date, 1, 7)」で変換できます

実践SQL問題④

難易度	★★☆☆☆
所要時間	5分

問題

月別の売上と購入件数を集計してください。ただし注文キャンセルのデータは除外してください。

ヒント①	「月」情報の取得はsubstrを使って日付から「月」に変換しましょう
ヒント②	「月」情報は「substr(order_date, 1, 7)」で変換できます
ヒント③	月でGROUP BYして売上と購入件数を集計しましょう

実践SQL問題④　解答例

難易度	★★☆☆☆
所要時間	5分

```sql
SELECT
    substr(o.order_date, 1, 7) AS order_month,
    COUNT(*),
    SUM(p.price)
FROM orders AS o
LEFT JOIN products AS p ON o.order_product_id = p.product_id
WHERE o.is_canceled = 0
GROUP BY order_month
ORDER BY order_month
```

実践SQL問題④　解答例

難易度	★★☆☆☆
所要時間	5分

```sql
SELECT
    substr(o.order_date, 1, 7) AS order_month,
    COUNT(*),
    SUM(p.price)
FROM orders AS o
LEFT JOIN products AS p ON o.order_product_id = p.product_id
WHERE o.is_canceled = 0
GROUP BY order_month
ORDER BY order_month
```

- 「月」はsubstr(order_date, 1, 7)を使って取得

order_date	order_month
2020-01-01	2020-01
	substr(order_date, 1, 7)

日付の1文字目から7文字を切り取って「月」の情報を取得

実践SQL問題④　解答例

難易度	★★☆☆☆
所要時間	5分

```sql
SELECT
    substr(o.order_date, 1, 7) AS order_month,
    COUNT(*),
    SUM(p.price)
FROM orders AS o
LEFT JOIN products AS p ON o.order_product_id = p.product_id
WHERE o.is_canceled = 0
GROUP BY order_month
ORDER BY order_month
```

- SELECT内でsubstrで変換したorder_monthを使ってGROUP BYとORDER BYで集計と並び替え
- 本来であればSQLの実行順からORDER BYではSELECTの別名が使えるが、GROUP BYでは使えない
- DBによってGROUP BYでもSELECTの別名が使える場合があるので注意が必要

SQLの実行順序 →

① FROM
② JOIN
③ WHERE
④ GROUP BY
⑤ HAVING
⑥ SELECT
⑦ ORDER BY
⑧ LIMIT

実践SQL問題④　解答例（別）

難易度	★★☆☆☆
所要時間	5分

```sql
WITH order_data_month AS(
    SELECT
        *,
        substr(order_date, 1, 7) AS order_month
    FROM orders
    WHERE is_canceled = 0
)

SELECT
    order_month,
    COUNT(*),
    SUM(p.price)
FROM order_data_month AS o
LEFT JOIN products AS p ON o.order_product_id = p.product_id
GROUP BY order_month
ORDER BY order_month
```

- WITH句を使って、「月」情報を追加した一時テーブルを作成
- 月情報でGROUP BYを使って集計
- この場合はWITH句の中で定義したorder_monthを使ってGROUP BYの集計が可能

実践SQL問題集　10選

No	問題	難易度
1	性別ごとの人数を集計してください	★☆☆☆☆
2	年代別の人数を集計してください	★★★☆☆
3	日別の売上と購入件数を集計してください	★★☆☆☆
4	月別の売上と購入件数を集計してください	★★☆☆☆
5	ユーザーごとの最終購入日/購入頻度/購入金額を集計してください	★★★☆☆
6	購入頻度(注文日数)ごとにユーザー数を集計してください①	★★★★☆
7	購入頻度(注文日数)ごとにユーザー数を集計してください②	★★★★★
8	「食品」を購入したユーザーIDをユニーク(重複なし)で抽出してください	★★☆☆☆
9	「美容」を買っている人が他に何を買っているか抽出して下さい	★★★★☆
10	初回購入で何が一番購入されているか抽出して下さい	★★★★★

実践SQL問題⑤

難易度	★★★☆☆
所要時間	5分

問題

ユーザーごとの最終購入日、購入頻度(注文日数)、購入金額を集計してください(RFM分析)。ただし注文キャンセルのデータは除外して削除済ユーザー、未購入ユーザーも除外してください。

実践SQL問題⑤

難易度	★★★☆☆
所要時間	5分

問題

ユーザーごとの最終購入日、購入頻度(注文日数)、購入金額を集計してください(RFM分析)。ただし注文キャンセルのデータは除外して削除済ユーザー、未購入ユーザーも除外してください。

ヒント①

usersテーブルとproductsテーブルとJOINして2つテーブルを結合して
ユーザーごとの購入データを確認しましょう

実践SQL問題⑤

難易度	★★★☆☆
所要時間	5分

問題

ユーザーごとの最終購入日、購入頻度(注文日数)、購入金額を集計してください(RFM分析)。ただし注文キャンセルのデータは除外して削除済ユーザー、未購入ユーザーも除外してください。

ヒント①	usersテーブルとproductsテーブルとJOINして2つテーブルを結合してユーザーごとの購入データを確認しましょう
ヒント②	ユーザーごとの集計をする場合はuser_idでGROUP BYを使いましょう

実践SQL問題⑤

難易度	★★★☆☆
所要時間	5分

問題

ユーザーごとの最終購入日、購入頻度(注文日数)、購入金額を集計してください(RFM分析)。ただし注文キャンセルのデータは除外して削除済ユーザー、未購入ユーザーも除外してください。

ヒント①	usersテーブルとproductsテーブルとJOINして2つテーブルを結合してユーザーごとの購入データを確認しましょう
ヒント②	ユーザーごとの集計をする場合はuser_idでGROUP BYを使いましょう
ヒント③	・最終購入日 → 購入日の最大値 ・購入頻度(注文日数) → 購入日数のカウント　として計算してみましょう ・購入金額 → 金額の合計

実践SQL問題⑤　解答例

難易度	★★★☆☆
所要時間	5分

```sql
SELECT
    o.user_id,
    MAX(o.order_date),
    COUNT(DISTINCT o.order_date),
    SUM(p.price)
FROM orders o
LEFT JOIN products p ON o.order_product_id = p.product_id
LEFT JOIN users u ON o.user_id = u.user_id
WHERE o.is_canceled = 0
AND u.is_deleted = 0
GROUP BY o.user_id
```

実践SQL問題⑤　解答例

難易度	★★★☆☆
所要時間	5分

```sql
SELECT
    o.user_id,
    MAX(o.order_date),
    COUNT(DISTINCT o.order_date),
    SUM(p.price)
FROM orders o
LEFT JOIN products p ON o.order_product_id = p.product_id
LEFT JOIN users u ON o.user_id = u.user_id
WHERE o.is_canceled = 0
AND u.is_deleted = 0
GROUP BY o.user_id
```

- ordersとproductsテーブルをJOIN
  - 商品の金額（price）を取得するため
- ordersとusersテーブルをJOIN
  - 削除済みユーザーを除外するため

実践SQL問題⑤　解答例

難易度	★★★☆☆
所要時間	5分

```sql
SELECT
    o.user_id,
    MAX(o.order_date),
    COUNT(DISTINCT o.order_date),
    SUM(p.price)
FROM orders o
LEFT JOIN products p ON o.order_product_id = p.product_id
LEFT JOIN users u ON o.user_id = u.user_id
WHERE o.is_canceled = 0
AND u.is_deleted = 0
GROUP BY o.user_id
```

- user_idでGROUP BYして、以下でRFMを抽出
  - R：最終購入日→MAX(o.order_date)
  - F：購入日数→COUNT(DISTINCT o.order_date)
  - M：購入金額→SUM(p.price)

※RFM分析とはRecency（最終購入日）Frequency（購入頻度）Monetary（購入金額）の3つの指標で顧客を分析する手法

実践SQL問題集　10選

No	問題	難易度
1	性別ごとの人数を集計してください	★☆☆☆☆
2	年代別の人数を集計してください	★★★☆☆
3	日別の売上と購入件数を集計してください	★★☆☆☆
4	月別の売上と購入件数を集計してください	★★☆☆☆
5	ユーザーごとの最終購入日/購入頻度/購入金額を集計してください	★★★☆☆
6	購入頻度(注文日数)ごとにユーザー数を集計してください①	★★★★☆
7	購入頻度(注文日数)ごとにユーザー数を集計してください②	★★★★★
8	「食品」を購入したユーザーIDをユニーク(重複なし)で抽出してください	★★☆☆☆
9	「美容」を買っている人が他に何を買っているか抽出して下さい	★★★★☆
10	初回購入で何が一番購入されているか抽出して下さい	★★★★★

実践SQL問題⑥

難易度	★★★★☆
所要時間	5分

問題

年間の購入で何日間購入があったか購入頻度(注文日数)ごとにユーザー数を集計してください。
ただし注文キャンセルのデータは除外して削除済ユーザーと未購入ユーザーも除外してください。

アウトプットイメージ

購入日数	購入者数		
10	50	→	年間の購入日数が10日のユーザーが50人
20	150	→	年間の購入日数が20日のユーザーが150人

実践SQL問題⑥

難易度	★★★★☆
所要時間	5分

問題

年間の購入で何日間購入があったか購入頻度(注文日数)ごとにユーザー数を集計してください。
ただし注文キャンセルのデータは除外して削除済ユーザーと未購入ユーザーも除外してください。

ヒント①	ユーザーごとに購入頻度(購入日数)をカウントしましょう

実践SQL問題⑥

難易度	★★★★☆
所要時間	5分

問題

年間の購入で何日間購入があったか購入頻度(注文日数)ごとにユーザー数を集計してください。
ただし注文キャンセルのデータは除外して削除済ユーザーと未購入ユーザーも除外してください。

ヒント①	ユーザーごとに購入頻度(購入日数)をカウントしましょう
ヒント②	WITH句を使って一時テーブルを作ると整理しやすくなります(その際はSELECTした結果にASでカラム名をつけましょう)

実践SQL問題⑥

難易度	★★★★☆
所要時間	5分

問題

年間の購入で何日間購入があったか購入頻度(注文日数)ごとにユーザー数を集計してください。
ただし注文キャンセルのデータは除外して削除済ユーザーと未購入ユーザーも除外してください。

ヒント①	ユーザーごとに購入頻度(購入日数)をカウントしましょう
ヒント②	WITH句を使って一時テーブルを作ると整理しやすくなります(その際はSELECTした結果にASでカラム名をつけましょう)
ヒント③	購入頻度(注文日数)ごとにユーザー数をカウントする場合は購入頻度(注文日数)でGROUP BYを使いましょう

実践SQL問題⑥　解答例

難易度	★★★★☆
所要時間	5分

```sql
WITH order_data_users AS(
    SELECT
        o.user_id,
        COUNT(DISTINCT order_date) AS count_order_date
    FROM orders o
    LEFT JOIN products p ON o.order_product_id = p.product_id
    LEFT JOIN users u ON o.user_id = u.user_id
    WHERE is_canceled = 0
    AND u.is_deleted = 0
    AND order_date BETWEEN '2022-01-01' and '2022-12-31'
    GROUP BY o.user_id
)

SELECT
    count_order_date,
    COUNT(DISTINCT user_id) as user_count
FROM order_data_users
GROUP BY count_order_date
ORDER BY count_order_date
```

実践SQL問題⑥　解答例

難易度	★★★★☆
所要時間	5分

```sql
WITH order_data_users AS(
    SELECT
        o.user_id,
        COUNT(DISTINCT order_date) AS count_order_date
    FROM orders o
    LEFT JOIN products p ON o.order_product_id = p.product_id
    LEFT JOIN users u ON o.user_id = u.user_id
    WHERE is_canceled = 0
    AND u.is_deleted = 0
    AND order_date BETWEEN '2022-01-01' and '2022-12-31'
    GROUP BY o.user_id
)

SELECT
    count_order_date,
    COUNT(DISTINCT user_id) as user_count
FROM order_data_users
GROUP BY count_order_date
ORDER BY count_order_date
```

- WITH句を使ってまずはユーザーごとの購入日数をカウントする
- ordersとusersテーブルをJOINして注文データを抽出
- 条件でキャンセルを除外、削除済ユーザーの除外をする
- ordersからデータを抽出しているので、購入があったユーザーのみデータを抽出

実践SQL問題⑥　解答例

難易度	★★★★☆
所要時間	5分

```sql
WITH order_data_users AS(
    SELECT
        o.user_id,
        COUNT(DISTINCT order_date) AS count_order_date
    FROM orders o
    LEFT JOIN products p ON o.order_product_id = p.product_id
    LEFT JOIN users u ON o.user_id = u.user_id
    WHERE is_canceled = 0
    AND u.is_deleted = 0
    AND order_date BETWEEN '2022-01-01' and '2022-12-31'
    GROUP BY o.user_id
)

SELECT
    count_order_date,
    COUNT(DISTINCT user_id) as user_count
FROM order_data_users
GROUP BY count_order_date
ORDER BY count_order_date
```

- user_idでGROUP BYすることでユーザー毎の購入日数を集計
- 購入日数の集計はCOUNT(DISTINCT order_date）→購入日数をユニークカウントする

実践SQL問題⑥　解答例

難易度	★★★★☆
所要時間	5分

```sql
WITH order_data_users AS(
    SELECT
        o.user_id,
        COUNT(DISTINCT order_date) AS count_order_date
    FROM orders o
    LEFT JOIN products p ON o.order_product_id = p.product_id
    LEFT JOIN users u ON o.user_id = u.user_id
    WHERE is_canceled = 0
    AND u.is_deleted = 0
    AND order_date BETWEEN '2022-01-01' and '2022-12-31'
    GROUP BY o.user_id
)

SELECT
    count_order_date,
    COUNT(DISTINCT user_id) as user_count
FROM order_data_users
GROUP BY count_order_date
ORDER BY count_order_date
```

- ユーザーごとの購入日数をカウントした一時テーブルから購入日数をベースにしてGROUP BYで人数をカウント

実践SQL問題⑥　解説

まずはユーザー毎の購入日数を集計し、その後購入日数でユーザー数を集計する

STEP 1

現状のテーブルの確認

usersテーブル

ordersテーブル

STEP 2

ユーザー毎の購入日数を集計

user_id	count_order_date
A0001	100
A0002	30
A0003	30
A0004	80

STEP 3

購入日数でユーザー数を集計

count_order_date	user_count
30	2
80	1
100	1

実践SQL問題集　10選

No	問題	難易度
1	性別ごとの人数を集計してください	★☆☆☆☆
2	年代別の人数を集計してください	★★★☆☆
3	日別の売上と購入件数を集計してください	★★☆☆☆
4	月別の売上と購入件数を集計してください	★★☆☆☆
5	ユーザーごとの最終購入日/購入頻度/購入金額を集計してください	★★★☆☆
6	購入頻度(注文日数)ごとにユーザー数を集計してください①	★★★★☆
7	**購入頻度(注文日数)ごとにユーザー数を集計してください②**	**★★★★★**
8	「食品」を購入したユーザーIDをユニーク(重複なし)で抽出してください	★★☆☆☆
9	「美容」を買っている人が他に何を買っているか抽出して下さい	★★★★☆
10	初回購入で何が一番購入されているか抽出して下さい	★★★★★

実践SQL問題⑦

難易度	★★★★★
所要時間	5分

問題

年間の購入で何日間購入があったか購入頻度(注文日数)ごとにユーザー数を集計してください。
ただし未購入ユーザーも含めた購入日数をカウントして、注文キャンセルのデータは除外、削除済みユーザーは除外して下さい。

アウトプットイメージ

購入日数	購入者数		
0	100	→	年間の購入日数が0日のユーザーが100人 （未購入ユーザー）
20	150	→	年間の購入日数が20日のユーザーが150人

実践SQL問題⑦

難易度	★★★★★
所要時間	5分

問題

年間の購入で何日間購入があったか購入頻度(注文日数)ごとにユーザー数を集計してください。
ただし未購入ユーザーも含めた購入日数をカウントして、注文キャンセルのデータは除外、削除済みユーザーは除外して下さい。

ヒント①

考え方は問題⑥と同じでまずはユーザーごとの購入日数をカウントしましょう

実践SQL問題⑦

難易度	★★★★★
所要時間	5分

問題

年間の購入で何日間購入があったか購入頻度(注文日数)ごとにユーザー数を集計してください。
ただし未購入ユーザーも含めた購入日数をカウントして、注文キャンセルのデータは除外、削除済みユーザーは除外して下さい。

ヒント①	考え方は問題⑥と同じでまずはユーザーごとの購入日数をカウントしましょう
ヒント②	購入がなかったユーザーもカウントするので、usersをベースにデータを取得しましょう

実践SQL問題⑦

難易度	★★★★★
所要時間	5分

問題

年間の購入で何日間購入があったか購入頻度(注文日数)ごとにユーザー数を集計してください。
ただし未購入ユーザーも含めた購入日数をカウントして、注文キャンセルのデータは除外、削除済みユーザーは除外して下さい。

ヒント①	考え方は問題⑥と同じでまずはユーザーごとの購入日数をカウントしましょう
ヒント②	購入がなかったユーザーもカウントするので、usersをベースにデータを取得しましょう
ヒント③	購入頻度(注文日数)ごとにユーザー数をカウントする場合は購入頻度(注文日数)でGROUP BYを使いましょう

実践SQL問題⑦　解答例

難易度	★★★★★
所要時間	5分

```sql
WITH order_data_users AS(
    SELECT
        u.user_id,
        COUNT(DISTINCT order_date) AS count_order_date
    FROM users u
    LEFT JOIN orders o ON u.user_id = o.user_id
    WHERE (o.is_canceled = 0 OR o.is_canceled IS NULL)
    AND u.is_deleted = 0
    GROUP BY u.user_id
)

SELECT
    count_order_date,
    COUNT(DISTINCT user_id) AS user_count
FROM order_data_users
GROUP BY count_order_date
ORDER BY count_order_date
```

実践SQL問題⑦　解答例

難易度	★★★★★
所要時間	5分

```sql
WITH order_data_users AS(
    SELECT
        u.user_id,
        COUNT(DISTINCT order_date) AS count_order_date
    FROM users u
    LEFT JOIN orders o ON u.user_id = o.user_id
    WHERE (o.is_canceled = 0 OR o.is_canceled IS NULL)
    AND u.is_deleted = 0
    GROUP BY u.user_id
)

SELECT
    count_order_date,
    COUNT(DISTINCT user_id) AS user_count
FROM order_data_users
GROUP BY count_order_date
ORDER BY count_order_date
```

- WITH句を使ってまずはユーザーごとの購入日数をカウントする
- 購入がなかったユーザーもカウントするので、ベースはusersテーブルから取得する。usersテーブルに対してordersをJOINさせる
- キャンセルを除外する際にis_canceled = 0だけにすると、そもそも購入がなかったデータが全て除外されてしまう(購入がない場合はis_canceledがnull)なので、is_canceledがnullの場合も条件に含める

実践SQL問題⑦　解答例

難易度	★★★★★
所要時間	5分

```sql
WITH order_data_users AS(
    SELECT
        u.user_id,
        COUNT(DISTINCT order_date) AS count_order_date
    FROM users u
    LEFT JOIN orders o ON u.user_id = o.user_id
    WHERE (o.is_canceled = 0 OR o.is_canceled IS NULL)
    AND u.is_deleted = 0
    GROUP BY u.user_id
)

SELECT
    count_order_date,
    COUNT(DISTINCT user_id) AS user_count
FROM order_data_users
GROUP BY count_order_date
ORDER BY count_order_date
```

- ⑥との違いはJOINの順番
- ⑥はordersをベースにしてusersとJOIN。今回はusersをベースにしてordersとJOIN

実践SQL問題⑦　解答例

難易度　★★★★★

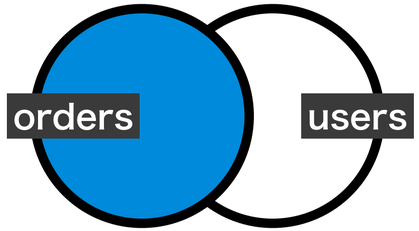
所要時間　5分

```sql
WITH order_data_users AS(
    SELECT
        u.user_id,
        COUNT(DISTINCT order_date) AS count_order_date
    FROM users u
    LEFT JOIN orders o ON u.user_id = o.user_id
    WHERE (o.is_canceled = 0 OR o.is_canceled IS NULL)
    AND u.is_deleted = 0
    GROUP BY u.user_id
)

SELECT
    count_order_date,
    COUNT(DISTINCT user_id) AS user_count
FROM order_data_users
GROUP BY count_order_date
ORDER BY count_order_date
```

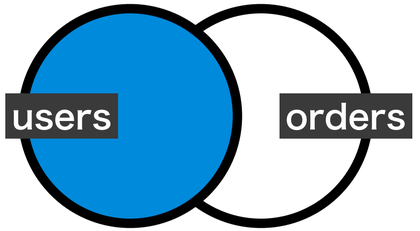
- ⑥との違いはJOINの順番
- ⑥はordersをベースにしてusersとJOIN。今回はusersをベースにしてordersとJOIN

問題⑥
ordersをベースにusersとJOIN



・ordersのデータが全て結合される
→usersに情報がない購入データ（未ログインで購入したデータ）も取得される
・購入がないデータは取得されない

問題⑦
usersをベースにordersとJOIN



・usersのデータが全て結合される
→ordersに情報がない顧客情報（未購入のユーザー）も取得される
・usersに情報がない購入データ（未ログインで購入したデータ）はは取得されない

実践SQL問題⑦　解答例

難易度	★★★★★
所要時間	5分

```sql
WITH order_data_users AS(
    SELECT
        u.user_id,
        COUNT(DISTINCT order_date) AS count_order_date
    FROM users u
    LEFT JOIN orders o ON u.user_id = o.user_id
    WHERE (o.is_canceled = 0 OR o.is_canceled IS NULL)
    AND u.is_deleted = 0
    GROUP BY u.user_id
)

SELECT
    count_order_date,
    COUNT(DISTINCT user_id) AS user_count
FROM order_data_users
GROUP BY count_order_date
ORDER BY count_order_date
```

- あとは問題⑥同様に購入頻度でGROUP BYをしてユーザー数をカウントする

実践SQL問題集　10選

No	問題	難易度
1	性別ごとの人数を集計してください	★☆☆☆☆
2	年代別の人数を集計してください	★★★☆☆
3	日別の売上と購入件数を集計してください	★★☆☆☆
4	月別の売上と購入件数を集計してください	★★☆☆☆
5	ユーザーごとの最終購入日/購入頻度/購入金額を集計してください	★★★☆☆
6	購入頻度(注文日数)ごとにユーザー数を集計してください①	★★★★☆
7	購入頻度(注文日数)ごとにユーザー数を集計してください②	★★★★★
8	「食品」を購入したユーザーIDをユニーク(重複なし)で抽出してください	★★☆☆☆
9	「美容」を買っている人が他に何を買っているか抽出して下さい	★★★★☆
10	初回購入で何が一番購入されているか抽出して下さい	★★★★★

実践SQL問題⑧

難易度	★★☆☆☆
所要時間	5分

問題

2022年1月1日に「食品」を購入したユーザーIDをユニーク(重複なし)で抽出してください。ただし注文キャンセルのデータは除外してください。

実践SQL問題⑧

難易度	★★☆☆☆
所要時間	5分

問題

2022年1月1日に「食品」を購入したユーザーIDをユニーク(重複なし)で抽出してください。ただし注文キャンセルのデータは除外してください。

ヒント①	注文情報と商品情報を結合しましょう

実践SQL問題⑧

難易度	★★☆☆☆
所要時間	5分

問題

2022年1月1日に「食品」を購入したユーザーIDをユニーク(重複なし)で抽出してください。ただし注文キャンセルのデータは除外してください。

ヒント①	注文情報と商品情報を結合しましょう
ヒント②	複数の条件をつける場合はWHERE〜ANDを使いましょう

実践SQL問題⑧

難易度	★★☆☆☆
所要時間	5分

問題

2022年1月1日に「食品」を購入したユーザーIDをユニーク(重複なし)で抽出してください。ただし注文キャンセルのデータは除外してください。

ヒント①	注文情報と商品情報を結合しましょう
ヒント②	複数の条件をつける場合はWHERE〜ANDを使いましょう
ヒント③	「食品」はlarge_categoryで判定しましょう

実践SQL問題⑧　解答例

難易度	★★☆☆☆
所要時間	5分

```sql
SELECT
    DISTINCT o.user_id
FROM orders o
LEFT JOIN products p ON o.order_product_id = p.product_id
WHERE is_canceled = 0
AND o.order_date = '2022-01-01'
AND p.large_category = '食品'
```

実践SQL問題⑧　解答例

難易度	★★☆☆☆
所要時間	5分

```sql
SELECT
    DISTINCT o.user_id
FROM orders o
LEFT JOIN products p ON o.order_product_id = p.product_id
WHERE is_canceled = 0
AND o.order_date = '2022-01-01'
AND p.large_category = '食品'
```

- ordersとproductsをproduct_idをキーに結合します
- オーダーキャンセルの情報は除外します
- 「食品」はlarge_category = ‘食品’で判定します

実践SQL問題⑧　解答例　TIPS

```
SELECT
  DISTINCT large_category
FROM products
```

出力結果

large_category
食品
日用品
インテリア
ファッション
電化製品

- large_categoryにどんな値が入っているのか確認したい場合、「DISTINCT」を使ってlarge_categoryのユニークな値を一覧で確認する事ができます
- これでlarge_categoryに「食品」のデータが入っている事が確認できます(データ分析の際によく確認します)

実践SQL問題集　10選

No	問題	難易度
1	性別ごとの人数を集計してください	★☆☆☆☆
2	年代別の人数を集計してください	★★★☆☆
3	日別の売上と購入件数を集計してください	★★☆☆☆
4	月別の売上と購入件数を集計してください	★★☆☆☆
5	ユーザーごとの最終購入日/購入頻度/購入金額を集計してください	★★★☆☆
6	購入頻度(注文日数)ごとにユーザー数を集計してください①	★★★★☆
7	購入頻度(注文日数)ごとにユーザー数を集計してください②	★★★★★
8	「食品」を購入したユーザーIDをユニーク(重複なし)で抽出してください	★★☆☆☆
9	「美容」を買っている人が他に何を買っているか抽出して下さい	★★★★☆
10	初回購入で何が一番購入されているか抽出して下さい	★★★★★

実践SQL問題⑨

難易度	★★★★☆
所要時間	5分

問題

「美容」を買っている人が他に何を買っているか、カテゴリごとに購入者数を抽出して下さい。ただし注文キャンセルのデータは除外してください。また中カテゴリ(medium_category)ごとの購入者が多い順に結果を出力してください。

実践SQL問題⑨

難易度	★★★★☆
所要時間	5分

問題

「美容」を買っている人が他に何を買っているか、カテゴリごとに購入者数を抽出して下さい。ただし注文キャンセルのデータは除外してください。また中カテゴリ(medium_category)ごとの購入者が多い順に結果を出力してください。

ヒント①

WITH句を使って「美容を買った人」の一時テーブルを作成しましょう

実践SQL問題⑨

難易度	★★★★☆
所要時間	5分

問題

「美容」を買っている人が他に何を買っているか、カテゴリごとに購入者数を抽出して下さい。ただし注文キャンセルのデータは除外してください。また中カテゴリ(medium_category)ごとの購入者が多い順に結果を出力してください。

ヒント①	WITH句を使って「美容を買った人」の一時テーブルを作成しましょう
ヒント②	「美容を買った人」AND「美容以外の商品」と2つの条件を付けてデータを取得しましょう

実践SQL問題⑨

難易度	★★★★☆
所要時間	5分

問題

「美容」を買っている人が他に何を買っているか、カテゴリごとに購入者数を抽出して下さい。ただし注文キャンセルのデータは除外してください。また中カテゴリ(medium_category)ごとの購入者が多い順に結果を出力してください。

ヒント①	WITH句を使って「美容を買った人」の一時テーブルを作成しましょう
ヒント②	「美容を買った人」AND「美容以外の商品」と2つの条件を付けてデータを取得しましょう
ヒント③	最終的にはmedium_categoryでGROUP BYして購入者(user_id)をカウントしましょう

実践SQL問題⑨　解答例

難易度	★★★★☆
所要時間	5分

```sql
WITH beauty_order_user_id AS (
    SELECT
     DISTINCT user_id
    FROM orders o
    LEFT JOIN products p ON o.order_product_id = p.product_id
    WHERE is_canceled = 0
    AND p.medium_category = '美容'
)

SELECT
    p.medium_category,
    COUNT(DISTINCT user_id) AS uu
FROM orders o
LEFT JOIN products p ON o.order_product_id = p.product_id
WHERE is_canceled = 0
-- 美容を買った人
AND user_id IN (SELECT user_id FROM beauty_order_user_id)
-- 美容以外の商品
AND p.medium_category <> '美容'
GROUP BY p.medium_category
ORDER BY uu DESC
```

実践SQL問題⑨　解答例

難易度　★★★★☆

所要時間　5分

```sql
WITH beauty_order_user_id AS (
    SELECT
     DISTINCT user_id
    FROM orders o
    LEFT JOIN products p ON o.order_product_id = p.product_id
    WHERE is_canceled = 0
    AND p.medium_category = '美容'
)

SELECT
    p.medium_category,
    COUNT(DISTINCT user_id) AS uu
FROM orders o
LEFT JOIN products p ON o.order_product_id = p.product_id
WHERE is_canceled = 0
-- 美容を買った人
AND user_id IN (SELECT user_id FROM beauty_order_user_id)
-- 美容以外の商品
AND p.medium_category <> '美容'
GROUP BY p.medium_category
ORDER BY uu DESC
```

- WITH句を使って美容を買ったuser_idをユニークに抽出(問題⑧のSQLとほぼ同じ)

実践SQL問題⑨　解答例

難易度	★★★★☆
所要時間	5分

```sql
WITH beauty_order_user_id AS (
    SELECT
     DISTINCT user_id
    FROM orders o
    LEFT JOIN products p ON o.order_product_id = p.product_id
    WHERE is_canceled = 0
    AND p.medium_category = '美容'
)

SELECT
    p.medium_category,
    COUNT(DISTINCT user_id) AS uu
FROM orders o
LEFT JOIN products p ON o.order_product_id = p.product_id
WHERE is_canceled = 0
-- 美容を買った人
AND user_id IN (SELECT user_id FROM beauty_order_user_id)
-- 美容以外の商品
AND p.medium_category <> '美容'
GROUP BY p.medium_category
ORDER BY uu DESC
```

- 条件に「美容を買った人」をつける
  - user_idが美容を買ったユーザーIDに含まれる（IN）という条件をつける
- 美容以外の商品を抽出する場合は「<>」を使って特定のカテゴリ以外のデータを抽出する

実践SQL問題⑨　解答例

難易度	★★★★☆
所要時間	5分

```sql
WITH beauty_order_user_id AS (
    SELECT
     DISTINCT user_id
    FROM orders o
    LEFT JOIN products p ON o.order_product_id = p.product_id
    WHERE is_canceled = 0
    AND p.medium_category = '美容'
)

SELECT
    p.medium_category,
    COUNT(DISTINCT user_id) AS uu
FROM orders o
LEFT JOIN products p ON o.order_product_id = p.product_id
WHERE is_canceled = 0
-- 美容を買った人
AND user_id IN (SELECT user_id FROM beauty_order_user_id)
-- 美容以外の商品
AND p.medium_category <> '美容'
GROUP BY p.medium_category
ORDER BY uu DESC
```

- medium_categoryでGROUP BYしてuser_idをユニークカウントする

実践SQL問題⑨　解答例＋α

難易度	★★★★☆
所要時間	5分

```sql
WITH beauty_order_user_id AS (
    SELECT
        DISTINCT user_id
    FROM orders o
    LEFT JOIN products p ON o.order_product_id = p.product_id
    WHERE is_canceled = 0
    AND p.medium_category = '美容'
),

category_order_data AS (
    SELECT
        p.medium_category,
        COUNT(DISTINCT user_id) AS uu,
        (SELECT COUNT(DISTINCT user_id) FROM beauty_order_user_id) AS total_uu
    FROM orders o
    LEFT JOIN products p ON o.order_product_id = p.product_id
    WHERE is_canceled = 0
    -- 美容を買った人
    AND user_id IN (SELECT user_id FROM beauty_order_user_id)
    -- 美容以外の商品
    AND p.medium_category <> '美容'
    GROUP BY p.medium_category
)

SELECT
    medium_category,
    uu,
    total_uu,
    round(100.00 * uu / total_uu, 2) AS percent_uu
FROM category_order_data
ORDER BY uu DESC
```

他のカテゴリ買ってる人が何%いるのか割合も一緒に出す場合

medium_category	uu	total_uu	percent_uu
文房具	100	100	100%
飲料水	60	100	60%
野菜	50	100	50%

- サブクエリを使って、美容を買った人の合計人数を列に追加

実践SQL問題⑨　解答例＋α

難易度	★★★★☆
所要時間	5分

```sql
WITH beauty_order_user_id AS (
    SELECT
        DISTINCT user_id
    FROM orders o
    LEFT JOIN products p ON o.order_product_id = p.product_id
    WHERE is_canceled = 0
    AND p.medium_category = '美容'
),

category_order_data AS (
    SELECT
        p.medium_category,
        COUNT(DISTINCT user_id) AS uu,
        (SELECT COUNT(DISTINCT user_id) FROM beauty_order_user_id) AS total_uu
    FROM orders o
    LEFT JOIN products p ON o.order_product_id = p.product_id
    WHERE is_canceled = 0
    -- 美容を買った人
    AND user_id IN (SELECT user_id FROM beauty_order_user_id)
    -- 美容以外の商品
    AND p.medium_category <> '美容'
    GROUP BY p.medium_category
)

SELECT
    medium_category,
    uu,
    total_uu,
    round(100.00 * uu / total_uu, 2) AS percent_uu
FROM category_order_data
ORDER BY uu DESC
```

他のカテゴリ買ってる人が何%いるのか割合も一緒に出す場合

medium_category	uu	total_uu	percent_uu
文房具	100	100	100%
飲料水	60	100	60%
野菜	50	100	50%

- 合計人数とカテゴリごとの購入人数を割り算して割合を求める
- ROUNDは桁数を指定して四捨五入する関数
- 「100.00」をかけることで結果も小数点第二位まで表示させる

実践SQL問題集　10選

No	問題	難易度
1	性別ごとの人数を集計してください	★☆☆☆☆
2	年代別の人数を集計してください	★★★☆☆
3	日別の売上と購入件数を集計してください	★★☆☆☆
4	月別の売上と購入件数を集計してください	★★☆☆☆
5	ユーザーごとの最終購入日/購入頻度/購入金額を集計してください	★★★☆☆
6	購入頻度(注文日数)ごとにユーザー数を集計してください①	★★★★☆
7	購入頻度(注文日数)ごとにユーザー数を集計してください②	★★★★★
8	「食品」を購入したユーザーIDをユニーク(重複なし)で抽出してください	★★☆☆☆
9	「美容」を買っている人が他に何を買っているか抽出して下さい	★★★★☆
10	初回購入で何が一番購入されているか抽出して下さい	★★★★★

実践SQL問題⑩

難易度	★★★★★
所要時間	5分

問題

初回購入で何が一番購入されているか商品別のランキング(購入件数が多い順)を抽出して下さい。ただし注文キャンセルのデータは除外して、user_idが空のデータも除外してください。

実践SQL問題⑩

難易度	★★★★★
所要時間	5分

問題

初回購入で何が一番購入されているか商品別のランキング(購入件数が多い順)を抽出して下さい。ただし注文キャンセルのデータは除外して、user_idが空のデータも除外してください。

ヒント①

ユーザーごとの初回注文IDを抽出して、その注文IDのデータを抽出すれば
初回購入の購入データが抽出できます

実践SQL問題⑩

難易度	★★★★★
所要時間	5分

問題

初回購入で何が一番購入されているか商品別のランキング(購入件数が多い順)を抽出して下さい。ただし注文キャンセルのデータは除外して、user_idが空のデータも除外してください。

ヒント①	ユーザーごとの初回注文IDを抽出して、その注文IDのデータを抽出すれば初回購入の購入データが抽出できます
ヒント②	「初回購入」＝「ユーザー単位で注文番号が一番小さい注文」

実践SQL問題⑩

難易度	★★★★★
所要時間	5分

問題

初回購入で何が一番購入されているか商品別のランキング(購入件数が多い順)を抽出して下さい。ただし注文キャンセルのデータは除外して、user_idが空のデータも除外してください。

ヒント①	ユーザーごとの初回注文IDを抽出して、その注文IDのデータを抽出すれば初回購入の購入データが抽出できます
ヒント②	「初回購入」＝「ユーザー単位で注文番号が一番小さい注文」
ヒント③	「注文番号が一番小さい」を判定するためにMIN関数を使いましょう

実践SQL問題⑩　解答例

難易度	★★★★★
所要時間	5分

```sql
-- ユーザーごとの初回注文IDを抽出
WITH user_order AS (
    SELECT
        user_id,
        MIN(order_id) AS min_order_id
    FROM orders
    WHERE user_id IS NOT NULL
    AND is_canceled = 0
    GROUP BY user_id
)

SELECT
   name,
   COUNT(*) AS order_count,
   SUM(price) AS order_price
FROM orders AS o
LEFT JOIN products p ON o.order_product_id = p.product_id
-- 初回購入の注文IDに限定
WHERE order_id IN (SELECT min_order_id FROM user_order)
GROUP BY name
ORDER BY order_count DESC
```

実践SQL問題⑩　解答例

難易度	★★★★★
所要時間	5分

```sql
-- ユーザーごとの初回注文IDを抽出
WITH user_order AS (
    SELECT
        user_id,
        MIN(order_id) AS min_order_id
    FROM orders
    WHERE user_id IS NOT NULL
    AND is_canceled = 0
    GROUP BY user_id
)

SELECT
   name,
   COUNT(*) AS order_count,
   SUM(price) AS order_price
FROM orders AS o
LEFT JOIN products p ON o.order_product_id = p.product_id
-- 初回購入の注文IDに限定
WHERE order_id IN (SELECT min_order_id FROM user_order)
GROUP BY name
ORDER BY order_count DESC
```

- user_idでグルーピングして注文番号が一番小さいデータを抽出する一時テーブルを作成する
- 「MIN(order_id)」によってユーザーごとに一番小さい注文ID=初回の注文IDを取得する

実践SQL問題⑩　解答例

難易度	★★★★★
所要時間	5分

```sql
-- ユーザーごとの初回注文IDを抽出
WITH user_order AS (
    SELECT
        user_id,
        MIN(order_id) AS min_order_id
    FROM orders
    WHERE user_id IS NOT NULL
    AND is_canceled = 0
    GROUP BY user_id
)

SELECT
   name,
   COUNT(*) AS order_count,
   SUM(price) AS order_price
FROM orders AS o
LEFT JOIN products p ON o.order_product_id = p.product_id
-- 初回購入の注文IDに限定
WHERE order_id IN (SELECT min_order_id FROM user_order)
GROUP BY name
ORDER BY order_count DESC
```

- WITH句で作成した一時テーブルに初回購入の注文IDが含まれているので、それを条件に追加して初回購入のデータを抽出
- 「IN」を使って初回購入の注文IDのみを抽出

実践SQL問題⑩　解答例

難易度	★★★★★
所要時間	5分

```sql
-- ユーザーごとの初回注文IDを抽出
WITH user_order AS (
    SELECT
        user_id,
        MIN(order_id) AS min_order_id
    FROM orders
    WHERE user_id IS NOT NULL
    AND is_canceled = 0
    GROUP BY user_id
)

SELECT
   name,
   COUNT(*) AS order_count,
   SUM(price) AS order_price
FROM orders AS o
LEFT JOIN products p ON o.order_product_id = p.product_id
-- 初回購入の注文IDに限定
WHERE order_id IN (SELECT min_order_id FROM user_order)
GROUP BY name
ORDER BY order_count DESC
```

- 商品名（name）でグルーピングして購入件数を抽出(必要に応じて金額も出せる)
- 購入件数は「COUNT(*)」でも「COUNT(DISTINCT order_id)」でも同じ

実践SQL問題⑩　解答例（別解）

難易度	★★★★★
所要時間	5分

```sql
-- ユーザーごとの注文順番を追加
WITH add_order_number AS(
    SELECT
        *,
        DENSE_RANK() OVER(PARTITION BY o.user_id ORDER BY o.order_id ASC) AS order_number
    FROM orders AS o
    LEFT JOIN products p ON o.order_product_id = p.product_id
    WHERE o.user_id IS NOT NULL
    AND o.is_canceled = 0
)
SELECT
    name,
    COUNT(*) AS order_count,
    SUM(price) AS order_price
FROM add_order_number
WHERE order_number = 1
GROUP BY name
ORDER BY order_count DESC
```

RANK	同率があった場合順位は同じになり、その次は順位を飛ばします。 （1位、1位、3位・・・）
DENSE_RANK	同率があった場合順位は同じになり、その次は順位を飛ばしません。 （1位、1位、2位・・・）
ROW_NUMBER	同率があっても同じ順位にはならず、順位をカウントします。 同率があった場合の順位は常に同じとは限りません。 （1位、2位、3位・・・）

- WINDOW関数を使ってユーザーごとの注文に番号を付与する事で抽出が可能→DENSE_RANK()
- 初回購入のデータを抽出する場合は「order_number=1」の条件をつける

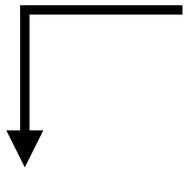
実践SQL問題⑩　WINDOW関数の解説

WINDOW関数は元のテーブルの行を維持したままGROUP BYで得られる結果（集約関数の結果）をテーブルに付加するイメージ

イメージ

元テーブル

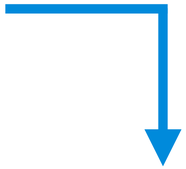
user_id	order_id	price
A0001	C00001	200
A0001	C00002	1000
A0002	C00003	500
A0002	C00004	800



GROUP BY

GROUP BYを使うとuser_idでグルーピングできるが行がuser_idごとにまとめられる

user_id	MIN(order_id)	SUM(price)
A0001	C00001	1200
A0002	C00003	1300



WINDOW関数

元のテーブルの行を維持したまま特定のカラムでグルーピングした結果を付与することができる

user_id	order_id	price	DENSE_RANK	SUM
A0001	C00001	200	1	1200
A0001	C00002	1000	2	1200
A0002	C00003	500	1	1300
A0002	C00004	800	2	1300

実践SQL問題⑩　WINDOW関数の解説

WINDOW関数は元のテーブルの行を維持したままGROUP BYで得られる結果（集約関数の結果）をテーブルに付加するイメージ

具体例

```
DENSE_RANK() OVER(PARTITION BY o.user_id ORDER BY o.order_id ASC) AS order_number
```

DENSE_RANK

『DENSE_RANK』は順番付をする関数。どのような順番にするかはOVER以下の記載内容に従う。
DENSE_RANK以外にもROW_NUMBERやSUMなどが使える。

OVER

『OVER』はWINDOW関数を使うという宣言。詳細はOVER以降に記載。

PARTITION BY

『PARTITION BY』はグループ条件を指定できる。GROUP BYと同じイメージ。

ORDER BY

『ORDER BY』は並び替えを指定（通常のORDER BYと同じ）

END