データベース 講義資料　第 4 回 データベース言語ＳＱＬ(1)

九州工業大学 情報工学部　講義担当：尾下真樹

1. データベース言語ＳＱＬ

リレーショナルデータベースにおけるデータベース操作のための言語として、SQL という言語が定義されている。
SQL では、リレーションを表（テーブル）として扱う。表は、リレーションと異なり、同一データの重複を許し、データの順番にも意味を持たせることができる。
SQL を用いて、データの問い合わせ（検索）、表の作成、表に対してのデータの挿入・削除・更新等の操作ができる。

2. 基本的な問い合わせの記述方法

SQL による問い合わせでは、リレーショナル代数式のように、どのような処理によりデータを出力するか（How）を記述するのではなく、どのようなデータを出力したいか（What）を記述する形式となっている。
基本的な記述方法としては、SELECT, FROM, WHERE, ORDER BY, GROUP BY, HAVING の各節 ＋ 節に関する記述、を組み合わせることにより記述する（図 1）。

```
SELECT　表.属性(値式), …
FROM　　表, …
WHERE　 条件式　AND …
```

– SELECT 節
　・問い合わせの結果として取り出す属性(値式)を指定
– FROM 節
　・どの表(テーブル)から検索するかを指定
– WHERE 節
　・検索の条件を指定
– ORDER BY節、GROUP BY節、HAVING節(後述)

図 1　SQL の基本的な記述方法

2.1. SELECT, FROM, WHERE 節の書き方

SELECT 節には、**表から取り出す属性（値式）を列挙**する。属性は、FROM 節に書いた表の属性である必要がある。
属性は、「表名.属性名」の形で記述するが、同一の名前の属性が一つしかない場合は、表名は省略できる。
FROM 節には、**表（テーブル）名を列挙**し、どの表から検索を行うかを指定する。複数の表を組み合わせて検索を行うときは、複数の表名を書ける（複数の表を用いる場合の方法は次回説明）。
WHERE 節には、**表から取り出すデータの条件を記述**する。FROM 節に書いた表のデータのうち、WHERE 節に書いた条件を満たすデータのみが、出力される。「AND」を使って、複数の条件を指定することもできる。この場合、全ての条件を満たすデータのみが、出力される。また、WHERE 節を省略すると、表の全てのデータが出力される。
WHERE 節に書く条件には、BETWEEN, IN, LIKE, NULL, EXISTS などの述語も使うことができる。述語は、属性名 BETWEEN x AND y（属性値が x～y の範囲に含まれるかを判定）、データ IN 表名（データが表に含まれるかを判定）、属性名 LIKE 文字列（属性の一部に文字列が含まれるか）、属性名 IS NULL（属性が空値か）、EXISTS 表名（表が空集合でないか）、のように用いることができる。

例えば、以下のようなリレーションがあるとする。（下線が引かれている属性は主キーを表す。）

　学生（学生番号、氏名）　科目（科目番号, 科目名, 単位数）　履修（科目番号, 学生番号, 成績）

このとき、以下の問い合わせ（質問）は、次のような SQL により記述できる。

　質問:　科目番号 001 の履修者の学生番号と成績の一覧を出力
　SQL:　SELECT　履修.学生番号, 履修.成績　FROM　履修　WHERE　履修.科目番号='001'

このとき、表名を省略して、以下のように書いても構わない。

　SQL:　SELECT　学生番号, 成績　FROM　履修　WHERE　科目番号= '001'

下記は、WHERE 節に AND や述語を使用する例である。

　質問:　学生番号 00100 の科目番号 005 の成績
　SQL:　SELECT　成績　FROM　履修　WHERE　学生番号='00100' AND 科目番号='005'

　質問:　科目番号 001 の科目で 80 点から 90 点の成績の学生番号の氏名の一覧を出力
　SQL:　SELECT　学生番号　FROM　履修　WHERE　科目番号='001' AND 成績 BETWEEN 80 AND 90

　質問:　名前に「子」の文字を含む学生の学生番号と氏名の一覧を出力
　SQL:　SELECT　学生番号, 氏名　FROM　学生　WHERE　氏名 LIKE '子'

2.2. SELECT 節の特殊な書き方（＊, DISTINCT, 集約演算）

SELECT 節の他の書き方として、＊（アスタリスク）を書くと、FROM 節に書いたテーブルの全ての属性が全て出力される。

質問: 単位数が 3 単位以上の科目の科目番号、科目名、単位数の一覧を出力
SQL: SELECT ＊ FROM 科目 WHERE 単位数>=3

SELECT 節に DISTINCT 指定をすると、出力するデータに重複がある場合には、重複が除去される。

質問: 全科目の科目名と単位の一覧を出力（同一の科目名のデータが複数ある場合には重複して出力しない）
SQL: SELECT DISTINCT 科目名, 単位数 FROM 科目

また、SELECT 節には集約関数を書ける。集約関数を使うと、それぞれのデータを出力する代わりに、全データの指定した属性に対して何らかの計算（集約関数）を適用し、演算結果を一つのデータ（一行のデータ）として出力する。集約関数には、COUNT（＊）… データの個数、SUM（属性）… 属性の合計値、AVG（属性）… 属性の平均値、MIN（属性）… 属性の最小値、MAX（属性）… 属性の最大値、がある。このうち、COUNT だけは、特定の属性についての関数ではないため、引数は常に＊とすることに注意する。

質問: 科目番号 001 の平均点、最小点、最高点の一覧を出力
SQL: SELECT AVG(成績), MIN(成績), MAX(成績) FROM 履修 WHERE 科目番号='001'

質問: 科目番号 001 の履修者の人数を出力
SQL: SELECT COUNT(＊) FROM 履修 WHERE 科目番号='001'
※ 履修データの個数が履修人数を表すと考えられるため、COUNT によりデータ数を出力する。

2.3. ORDER BY 節

ORDER BY 節には、データを出力する順番を決めるための属性を記述する。属性名の後ろに、ASC または、DESC をつけることで、属性値が小さいデータから大きいデータの昇順（ASC）で出力するか、その逆の降順（DESC）で出力するかを指定できる。

質問: 科目番号 005 を履修した学生の学生番号と成績の一覧を、成績の良い順番に出力
SQL: SELECT 学生番号, 成績 FROM 履修 WHERE 科目番号='005' ORDER BY 成績 DESC
※ 成績の良い順番とは、成績（点数）が大きいものから小さいものの順番であるため、降順（DESC）を使用。

2.4. GROUP BY + HAVING 節

GROUP BY 節には、データをグループにまとめるための属性を指定する。ここで指定した属性の値が等しいデータ同士がグループにまとめられて、グループごとのデータが出力される。
GROUP BY を指定した場合には、SELECT 節には集約関数、及び、GROUP BY に使用した属性しか書くことができない（普通の属性を指定しても、グループの中には複数のデータがあり、一つの属性値を出力できないため）。集約関数を使用する場合、通常は全データに対して集約関数が適用されるが、GROUP BY を行うことで、グループごとに集約関数が適用され、グループの数の分だけデータが出力されることになる。

質問: 全科目の科目番号と平均点の一覧を出力
SQL: SELECT 科目番号, AVG(成績) FROM 履修 GROUP BY 科目番号

HAVING 節に条件を記述すると、HAVING 節の条件を満たすグループのデータのみが出力される。HAVING 節にも、SELECT 節と同様、集約関数、及び、GROUP BY に使用した属性しか書くことができない。

質問: 履修者数が 30 名以上の科目の科目番号、履修者数、平均点の一覧を出力
SQL: SELECT 科目番号, COUNT(＊), AVG(成績) FROM 履修
GROUP BY 科目番号 HAVING COUNT(＊)>=30

なお、処理の適用順序としては、WHERE 節の条件は、GROUP BY＋HAVING 節によるグループ化よりも前に適用される。WHERE 節はデータに関する条件、HAVING 節にはグループに関する条件、を記述することに注意する。

質問: 各科目の、科目番号と 60 点以上の成績をとった履修者の数（※ 該当者が 0 名の科目は出力しなくとも良い）
SQL: SELECT 科目番号, COUNT(＊) FROM 履修 WHERE 成績>=60 GROUP BY 科目番号
※ この例では、最初に WHERE 節の条件によって成績が 60 以上のデータのみが取り出され、そのデータに対して、GROUP BY 節の属性によるグループ分けが適用される。