

便利な検索機能

- 集約演算
 - カウント、総和、平均、最大値、最小値
- 集約演算に関する便利な機能
 - Group-by
 - Having
- 出力に関する機能
 - Order-by

2019/4/25

データベース (©横田)

113

集約演算の例

```
SELECT COUNT(*)
FROM 製造元
WHERE 製造元="Hanon";
```

製造元	商品番号	商品名	製造元
P.CNT.ALL			Hanon

```
SELECT COUNT(DISTINCT 商品名)
FROM 製造元
WHERE 製造元="Hanon";
```

製造元	商品番号	商品名	製造元
		P.CNT.	Hanon

```
SELECT SUM(在庫)
FROM 秋葉原支店
WHERE 販売価格 ≤ 100,000;
```

秋葉原支店	商品番号	販売価格	在庫
		≤ 100,000	P.SUM.ALL

AVG(), MAX(), MIN()

AVG.ALL, MAX.ALL, MIN.ALL

2019/4/25

データベース (©横田)

114

Group-By演算の例

- 製造元毎の平均価格を求めたい

```
SELECT 製造元, 製造元 AS 製造元,
      AVG(秋葉原支店. 販売価格) AS 平均価格
FROM 製造元, 秋葉原支店
WHERE 製造元. 商品番号 = 秋葉原支店. 商品番号
GROUP BY 製造元, 製造元;
```

製造元	商品番号	商品名	製造元
	X		P.G.

秋葉原支店	商品番号	販売価格	在庫
	X	P.AVG.ALL	

結果

製造元	平均価格
GELL	120,000
Fepson	70,000
Hanon	30,000
Tenao	40,000

2019/4/25

データベース (©横田)

115

Having句 (Group-Byと共に)

SELECT 製造元, 製造元 **AS** 製造元
FROM 製造元, 秋葉原支店
WHERE 製造元, 商品番号=秋葉原支店, 商品番号
GROUP BY 製造元, 製造元
HAVING **AVG**(秋葉原支店, 販売価格) ≥ 50,000;

製造元	商品番号	商品名	製造元	CONDITION
	X		P.G.	AVG.ALL.Y ≥ 50,000
秋葉原支店	商品番号	販売価格	在庫	結果
	X	AVG.ALL.Y		
				製造元
				GELL
				Fepson

2019/4/25

データベース (©横田)

116

練習問題

1. 秋葉原支店において在庫数が 5 以下である商品の商品分類ごとの最低販売価格を示せ。
2. 秋葉原支店において在庫数が 5 以下である商品で最低販売価格が50,000 円以上の商品分類を示せ。

2019/4/25

データベース (©横田)

117

Order-By 句

昇順

SELECT *
FROM 秋葉原支店
WHERE 在庫 > 10
ORDER BY 販売価格 **ASC**;

秋葉原支店	商品番号	販売価格	在庫
P.		AO.	>10

降順

SELECT *
FROM 秋葉原支店
WHERE 在庫 > 10
ORDER BY 販売価格 **DESC**;

秋葉原支店	商品番号	販売価格	在庫
P.		DO.	>10

2019/4/25

データベース (©横田)

118

更新処理(Update)の例(1)

行単位挿入

INSERT INTO 製造元 (商品番号、商品名、製造元)
VALUES ("FIP08", "Inkjet Pt.", "Fepson");

製造元	商品番号	商品名	製造元
I.	FIP08	Inkjet Pt.	Fepson

検索結果挿入

INSERT INTO Fepson商品 (商品番号、商品名)
SELECT 商品番号、商品名
FROM 製造元
WHERE 製造元="Fepson";

製造元	商品番号	商品名	製造元	Fepson商品	商品番号	商品名
	X	Y	Fepson	I.	X	Y

2019/4/25

データベース (©横田)

119

更新処理(Update)の例(2)

検索結果による更新

UPDATE 秋葉原支店
SET 販売価格=秋葉原支店. 販売価格×0.8
WHERE 商品番号 **IN** (**SELECT** 商品番号
FROM 製造元
WHERE 製造元="Fepson");

製造元	商品番号	商品名	製造元
	X		Fepson

秋葉原支店	商品番号	販売価格	在庫
	X	Y	
	X	U.(0.8*Y)	

2019/4/25

データベース (©横田)

120

更新処理(Update)の例(3)

削除

DELETE
FROM 製造元
WHERE 製造元="Tenao";

製造元	商品番号	商品名	製造元
D.			Tenao

2019/4/25

データベース (©横田)

121

データ定義(制約)

- 一貫性制約 (Integrity Constraint)
 - 更新でデータベースの内容に矛盾が生じないようにする
 - キー制約 (Key Constraint)
 - キー属性に格納されるデータはnull値になってはならない
 - 外部キー[参照]制約 (Foreign Key [Referential] Constraint)
 - 外部からキーとして参照される値は消えてはならない

2019/4/25

データベース (©横田)

122

データ定義例(1)

- 関係の作成


```
CREATE TABLE 製造元
( 商品番号 CHAR(16) NOT NULL,
  商品名 CHAR(32) VAR,
  製造元 CHAR(16) VAR,
  希望小売価格 INTEGER,
  PRIMARY KEY(商品番号),
  FOREIGN KEY (商品番号) REFERENCES 秋葉原支店);
```

2019/4/25

データベース (©横田)

123

データ定義例(2)

- 関係の削除
 - DROP TABLE 秋葉原支店;
- 属性の追加／変更／削除
 - ALTER TABLE 製造元 ADD COLUMN 製造国 CHAR(8);
 - ALTER TABLE 製造元 ALTER COLUMN 製造元 SET 製造会社 CHAR(8);
 - ALTER TABLE 製造元 DROP COLUMN 希望小売価格;

2019/4/25

データベース (©横田)

124

埋め込み(組み込み) SQL

- 埋め込み SQL の文の先頭に EXEC SQL を付け、親言語の文と区別
- 親言語から変数で参照するため、: を変数の頭につける
- SQL の中では INTO 句を用いて親変数に代入
- 検索結果のタプルを1つずつアクセスするためにカーソルを使う

2019/4/25

データベース (©横田)

125

埋め込み(組み込み) SQL例(1)

```
EXEC SQL BEGIN DECLARE SECTION;
char item[16];
int price, amount;
EXEC SQL END DECLARE SECTION;
int x;
EXEC SQL CONNECT TO XXX;
EXEC SQL DECLARE c CURSOR FOR
SELECT ITEM, PRICE, AMOUNT
FROM Akihabara;
EXEC SQL OPEN c;
```

2019/4/25

データベース (©横田)

126

埋め込み(組み込み) SQL例(2)

```
EXEC SQL WHENEVER NOTFOUND GOTO printresult;
x = 0;
While(TRUE) {
    EXEC SQL FETCH c
        INTO :item, :price, :amount;
    if(price >= 100,000) x += amount;
}
printresult:
EXEC SQL CLOSE c;
printf("Total Amount = %d¥nl", x);
```

2019/4/25

データベース (©横田)

127