

SQL言語 (その2)

Database Systems, Xiangshi Ren, November 13, 2012

1

データ更新・問合せ

- データ更新
 - INSERT INTO (行の追加)
 - UPDATE (データ項目の更新)
 - DELETE (行の削除)
- 問合せ (クエリ、検索) : SELECT
 - 基本構文
 - 集合関数
 - 副問合せ

2

問合せSQL文 の4つの基本操作

- 二つ(または複数)の表を縦につなぐ→併合 (union)
- 二つ(または複数)の表を横につなぐ→結合(join) または直積(cartesian production)
- 行の取り出し→選択(selection)
- 列の取り出し→射影(projection)

3

データ操作: 問合せSQL文の基本構文

```
SELECT [ ALL | DISTINCT などの述語 ] 選択リスト
FROM テーブルリスト
[ WHERE 探索条件 ]
[ GROUP BY 列名リスト ]
[ HAVING 抽出条件 ]
[ ORDER BY 列名リスト [ ASC | DESC ] ];
```

[]はクエリの目的に必要であれば付け加える部分

4

SQL文の例

列 → SELECT 商品.商品名, 商品.単価
テーブル → FROM 商品
探索条件 → WHERE 商品.単価 > 200;

実行結果(ビュー)

	商品名	単価
▶	エリビール	¥280
	オタル白ラベル	¥300
	バードワイン	¥250
	ホワイトソルト	¥2,600
	ブラックペッパー	¥210

5

省略した書き方

```
SELECT 商品.商品名, 商品.単価
FROM 商品
WHERE 商品.単価 > 200;
```



```
SELECT 商品名, 単価
FROM 商品
WHERE 単価 > 200;
```

6

SELECT文の基本

キーワード	意味	使用例
AS	列名の別名定義	SELECT 列 AS A FROM 表
DISTINCT	ユニーク(一意)	SELECT DISTINCT 列 FROM 表
FROM	表指定	SELECT * FROM 表
WHERE	検索条件	SELECT * FROM 表 WHERE 列=100
GROUP BY	グループ化	SELECT 列 FROM 表 GROUP BY 列
ORDER BY	並べ替え	SELECT * FROM 表 ORDER BY 列 DESC(降順)
HAVING	グループ化後の絞り込み	SELECT 列 FROM 表 GROUP BY 列 HAVING 条件

8

GROUP BY句

SELECT 区分コード
FROM 商品;

SELECT 区分コード
FROM 商品
GROUP BY 区分コード;

商品区分
飲料
飲料
飲料
飲料
飲料
飲料
飲料
飲料
調味料
調味料
調味料

商品区分
飲料
調味料
菓子類
乳製品
穀類、シ
肉類
加工食品
魚介類

同じ区分コードのものをひとまとめにする

9

GROUP BY句の応用

SELECT 区分コード
FROM 商品;

SELECT 区分コード,
Count(*) AS 商品数
FROM 商品
GROUP BY 区分コード;

商品区分
飲料
飲料
飲料
飲料
飲料
飲料
飲料
調味料
調味料
調味料

テーブルの
最後まで数
えて16個

商品区分	商品数
飲料	16
調味料	18
菓子類	20
乳製品	14
穀類、シ	17
肉類	12
加工食品	14
魚介類	12

10

HAVING句

HAVING と GROUP BYは組で使う

SELECT 区分コード,
Count(*) AS 商品数
FROM 商品
GROUP BY 区分コード;

SELECT 区分コード,
Count(*) AS 商品数
FROM 商品
GROUP BY 区分コード
HAVING Count(*)>14;

商品区分	商品数
飲料	16
調味料	18
菓子類	20
乳製品	14
穀類、シ	17
肉類	12
加工食品	14
魚介類	12

商品区分	商品数
飲料	16
調味料	18
菓子類	20
穀類、シ	17

Count(*)>14

11

ORDER BY句

SELECT *
FROM 社員
ORDER BY 社員コード
[(ASC) | DESC];

社員コード	フリガナ	氏名	在籍支社
100	ショウモ	北門 基子	東京本社
105	モリウエ	森上 隆久馬	東京本社
106	タナカ	田中 和明	東京本社
107	カンラギ	喜城 孝史	東京本社
108	アキヤマ	秋山 秀樹	東京本社
110	カトウ	加藤 泰江	東京本社
203	トウライ	東海 望	大阪支社
204	カルラ	川村 匡	大阪支社
206	フルタ	古田 典典	大阪支社
207	マンザウ	松沢 真一	大阪支社

12

演算子

キーワード	意味	使用例
AND	論理積	SELECT * FROM 表 WHERE 列>10 AND 列<20
OR	論理和	SELECT * FROM 表 WHERE 列<10 OR 列>20
NOT	否定	SELECT * FROM 表 WHERE 列 NOT LIKE 'A%'('A%'は任意文字列のワイルドカード、Accessでは'*)
IN(NOT IN)	副問合わせ	SELECT * FROM 表1 WHERE 列 IN (SELECT 列 FROM 表2 WHERE 列=100)
EXISTS(NOT EXISTS)	存在すれば真	SELECT * FROM 表1 WHERE NOT EXISTS (SELECT * FROM 表2 WHERE 列=100)
BETWEEN	範囲抽出	SELECT * FROM 表 WHERE BETWEEN 10 AND 20
IS NULL	NULLを検出	SELECT * FROM 表 WHERE 列 IN NULL
LIKE	任意の文字	SELECT * FROM 表 WHERE 列 LIKE 'A%'('A%'は1文字のワイルドカード、Accessでは'*)

13

集合関数 (set function)

キーワード	意味	使用例
AVG	指定列の平均値	SELECT AVG(列) FROM 表
COUNT	結果の行数	SELECT COUNT(*) FROM 表
MAX	指定列の最大値	SELECT MAX(列) FROM 表
MIN	指定列の最小値	SELECT MIN(列) FROM 表
SUM	指定列の合計	SELECT SUM(列) FROM 表

14

『成績』

名前	科目	点数
鈴木	英語	80
川島	中国語	90
小泉	ドイツ語	55
川村	フランス語	65
高橋	英語	70
花田	フランス語	60
青木	ドイツ語	65
ISHIDA	フランス語	85
今村	英語	90

フランス語を履修した学生の数調べなさい。

フランス語の合計点を調べなさい。

英語の最低点を調べなさい。

ドイツ語の最高点を調べなさい。

15

複数の表を読む操作

SELECT 商品.区分.区分名, 商品.商品名
FROM 商品.区分, 商品
WHERE 商品.区分.区分コード=商品.区分コード;

商品

リンク

商品区分

商品コード	商品名	区分コード	区分コード	区分名	説明
1	果汁100%オレンジ	1	1	飲料	清涼飲料、コーヒー、紅茶、ビール、
2	果汁100%グレープ	1	2	調味料	甘味ソース、辛味ソース、香味、ス
3	果汁100%レモン	1	3	菓子類	デザート、キャンディー、菓子パン
4	果汁100%ピーチ	1	4	乳製品	チーズ
5	コーヒー・ココア・ミルク	1	5	穀類、シリアル	パン、クラッカー、パスタ、シリアル
6	コーヒー・ココア・ミルク	1	6	加工肉	加工肉
7	コーヒー・ココア・ミルク	1	7	加工食品	乾燥果物、豆腐
8	ビール・ビリアル・ビール	1	8	肉類	肉類
9	ビール・ビリアル・ビール	1	9	調味料	調味料
10	ビール・ビリアル・ビール	1	10	乳製品	乳製品

17

複数の表を読む操作(cont'd)

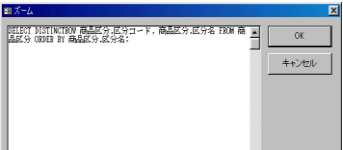
三つのテーブル

SELECT 受注.受注コード, 商品.商品名, 受注明細.単価
FROM 受注, 受注明細, 商品
WHERE 受注.受注コード=受注明細.受注コード
AND 受注明細.商品コード=商品.商品コード;



18

Ms-Accessのインタフェース (例2)

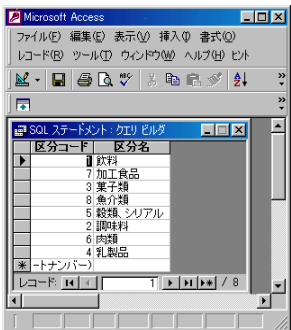


SELECT DISTINCT
商品区分. 区分コード, 商品区分.区分名
FROM 商品区分
ORDER BY 商品区分.区分名

商品区分テーブルから重複しない区分コードと区分名を選び、区分名の五十音順に並べる

19

SQLの実行結果



商品区分テーブルから、
区分コードと、
区分名が選択される

20

副問合せ

Select(Select.....);

21

問題:

「データベース」を履修した学生の名前を知りたいとする。

学生

学籍番号	名前
100	田中
200	小泉
300	金丸
400	岡崎
...	...

履修

学籍番号	クラス
100	英語
200	データベース
300	数学
400	データベース
...	...

22

問題:

「データベース」を履修した学生の名前を知りたいとする。

学籍番号が200と400の学生が「データベース」を履修していることを知っていれば,

(1)

```
SELECT 名前
FROM 学生
WHERE 学籍番号 IN [200,400];
```

学籍番号は前もって分からない場合,

(2)

```
SELECT 学籍番号
FROM 履修
WHERE クラス='データベース';
```



200
400

23

問題:

「データベース」を履修した学生の名前を知りたいとする。

(1)と(2)を組み合わせるにより,

```
SELECT 名前
FROM 学生
WHERE 学籍番号 IN (SELECT 学籍番号
FROM 履修
WHERE クラス='データベース');
```



小泉
岡崎

24

副問い合わせの構文

比較演算子 [ANY | ALL | SOME] (SELECT文)
式 [NOT] IN (SELECT文)
[NOT] EXISTS (SELECT文)

比較演算子 式、および、その式とサブクエリの結果とを比較する比較演算子。
式 サブクエリの結果の検索に使用する式。
SELECT文 書式や規則は他のSELECT文の場合と同じ。この SELECT文は括弧で囲む。

25

副問い合わせの例文

```
SELECT 氏名, 在籍支社, 入社日
FROM 社員
WHERE 入社日 <= ALL
(SELECT 入社日 FROM 社員 WHERE 在籍支社 = '東京本社');
```

氏名	在籍支社	入社日
森上 健久馬	東京本社	90/04/01
田中 和明	東京本社	90/04/01
加藤 泰江	東京本社	90/04/01
東海 金	大阪支社	89/04/01
川村 匡	大阪支社	90/04/01
新田 哲也	大阪支社	90/04/01
平野 彩子	大阪支社	88/04/01
高橋 良男	北九州支社	89/04/01
山本 雅治	北九州支社	89/04/01
香木 俊之	北九州支社	88/04/01
内野 義昭	北九州支社	87/04/01
小川 さよ子	北九州支社	89/04/01
稲垣 友子	北九州支社	90/04/01

<=ALL
すべてより
小さい

入社日
92/04/01
90/04/01
90/04/01
90/04/01
91/09/01
91/04/01
90/04/01

26

問題:
「売上」テーブルから、売上数の平均を下回っている
商品コードを検索せよ。

「売上」

商品コード	売上数
100	8
200	2
300	6
400	5
500	4
600	5

27

問題:
「売上」テーブルから、売上数の平均を下回っている商品
コードを検索せよ。

平均を求めるSELECT文は、
SELECT AVG(売上数)
FROM 売上;

⇒ 5

SELECT 商品コード
FROM 売上
WHERE 売上数 < 5;

SELECT 商品コード
FROM 売上
WHERE 売上数 < (SELECT AVG(売上数)
FROM 売上);

「売上」

商品コード	売上数
100	8
200	2
300	6
400	5
500	4
600	5

28

日付表記方法

SQL92

'年(4桁)-月(2桁)-日(2桁)'

Access 2000

#1999/09/23#

#99/09/23#

Oracle, SQL Server

'1999-09-23'

'99-09-23'

Visual dBASE

{10/02/97}

Approach

98/05/15

2000/05/15

PostgreSQL

'1999-09-23'

'19990923'

'09 23 1999'

'09-23-1999'

'1999/09/23'

'09/23/1999'

'Thu Sep 23 12:15:23 JST 1999'

'Thu Sep 23 1999'

'Sep 23 1999'

'1999 Sep 23'

29

文字列の表現(1)

SQL92, Oracle, SQLServer

	パターン	一致する文字列	一致しない文字列
任意数の文字	a%a	aa, aBa, aBBBa	aBC, aBaC
任意の1文字	a_a	aaa, a3a, aBa	aBBBa

30

文字列の表現(2)

Access

	パターン	一致する文字列	一致しない文字列
任意数の文字	a*a *ab* ab*	aa, aBa, aBBBa abC, aabb, Xab abYZ, abCDEF, abC	aBC acb, cadbe Cab, aab
任意の1文字	a?a	a0a, a3a, aBa	aBBBa
特殊文字	a[*]a	a*a	Aaa
1つの数字	a#a	a0a, a1a, a2a	aaa, a10a
指定範囲内の文字	[a-z]	f, p, j	2, &
指定範囲外の文字	[!a-z]	9, &, %	b, a
数字以外の文字	[!0-9]	A, a, &, ~	0, 1, 9
組み合わせ	a[!b-m]#	an9, az0, a99	abc, aj0

31