

部分問合せ

(入れ子型問合せ、nested query)

検索条件の中で別のselect文(**副問合せ(subquery)**と呼ぶ)を指定し、その結果を使って行う問合せ

- 副問合せを呼び出す方:**主問合せ(main query)**と呼ぶ
- 部分問合せには、以下のような種類がある。

比較演算子との組合せ

- 副問合せのselect文の結果が**単一の値**であれば、副問合せと比較演算子を組み合わせる

in述語の利用

- 副問合せのselect文の結果がある列の値の**集合**であるときは、副問合せとin述語を組み合わせる

この他、exists述語(主に、共通集合や差集合を求めるのに使用)があるが、共通集合・差集合は専用の集合演算があるので、exists述語は本講義では説明しない(「講義資料」には説明あり)

部分問合せの例(1)

比較演算子との組合せ

select * from emp
where empno =
(select manager from dept where dname = 'Account')

empno	ename	deptno	salary	roomno
E01	Smith	D01	100	R01

select * from emp
where empno = 'E01'
と等価

部分問合せの例(2)

in述語の利用

```
select * from emp
where deptno in
(select deptno from dept where dname = 'Account')
```

empno	ename	deptno	salary	roomno
E01	Smith	D01	100	R01
E02	Morgan	D01	70	R03
E03	Robert	D01	80	R02
E06	Taylor	D01	80	R12

- in述語は意味的には集合の要素の判定 ($a \in S$)
- 副問合せの結果の列に含まれない成分を持つタプルを求めたい時は **not in** を使う

結合問合せと部分問合せの 使い分け

- 一般に結合問合せの方が短く書ける
 - 少なくとも、結合問合せはselectを1回しか書かずに済む
- 単純に表を結合したいときには**結合問合せの方が使いやすい**
 - 表の結合であることが明確である
 - **外結合により結合結果でのタプルの欠損を防げる**
- それでは部分問合せは不要なのか？
 - 部分問合せでしか書けない問合せが存在する(後述)
 - 集約演算の結果との比較
 - リレーショナル代数の複数の演算の組合せ

SQLでの数値計算

SQLでは、他のプログラミング言語と同様、次のような数値計算を表現できる(列リストおよび検索条件中で数式として記述できる)

- 算術演算

+ (加算), - (減算), * (乗算), / (除算),

mod(x,y) (xをyで割った余り)

- 数値関数

ln(x) (xの自然対数)、log(x,y) (xを底とするyの対数)、

他に三角関数などがある

(例) `select empno, salary-20 from emp
where salary >= 100`

集約演算と数値関数(2)

- 何が問題なのか？
 - 集約演算は集約の対象を明示する必要があるため、問合せの**列リスト**でしか書けない
- 平均以下の給与をもらっている社員を求めたいときはどうすればよいか？
 - **部分問合せ**を使う必要がある

```
select empno, salary from emp  
where salary <=
```

```
(select avg(salary) from emp)
```

SQLによる空値の取り扱い(1)

- SQLの論理式は、真・偽以外に**未定義**の値を取る**3値論理**である
- 以下の2つの問合せの結果は同じにならないときがある
 - (問合せ1) `select count(*) from ref`
 - (問合せ2) `select count(*) from ref`
`where startpage<100 or startpage>=100`

なぜか？

SQLによる空値の取り扱い(2)

- 論理値が未定義となるのは、空値(NULL)が出てくるときである

id	...	startpage	endpage
R011		NULL	NULL

- 空値があると条件式の値が未定義となり、条件が成立しないと見なされる
- 空値かどうかを判定する、**x is null** (xが空値のとき真)、**x is not null** (xが空値のとき偽) という条件式がある

```
select count(*) from ref
```

```
where startpage is null or startpage<100 or  
startpage>=100
```

(結果は、select count(*) from refの結果と同じになる)

SQLによる集合演算

- **和両立**である2個の表RとSに対して次の問合せが可能

和集合演算($R \cup S$) <問合せ指定> union [all]<問合せ指定> [union [all]<問合せ指定>]

[例] select * from R **union** [all] select * from S

共通集合演算($R \cap S$) <問合せ指定> intersect <問合せ指定> [intersect <問合せ指定>]

[例] select * from R **intersect** select * from S

差集合演算($R - S$) <問合せ指定> except <問合せ指定> [except <問合せ指定>]

[例] select * from R **except** select * from S

(注) unionとunion allの違いは、集合演算の結果から重複したタプルを取り除くかどうか (unionだけだと取り除く)
差集合はOracleではexceptの代わりにminusで表記する

SQLによる集合演算(つづき)

- 任意の2つの表RとSに対して次の問合せが行える

直積演算($R \times S$)

```
select R.*, S.* from R,S
```

- ドメインが等しい列を持つ2個の表R(X,Y)とS(Y) ($\text{dom}(R.Y) = \text{dom}(S.Y)$)に対して次の問合せが行える(詳細は以降で説明)

商演算($R \div S$)

(参考)SQLの構文の意味

select 列リスト from 表リスト where 検索条件

- 実際の処理手順は from-where-selectの順で考えたほうが理解しやすい

例 select R.A from R,S where R.A=S.B

from R, S $R \times S$ (RとSの直積)

where R.A=S.B R.A=S.Bの条件で選択

select R.A R.Aのみを出力

(注意) select distinct R.Aとしない限り射影にはならない
(タプルの重複を取り除く必要があるため)

集合演算の例で使用する表

emp1 (salary \leq 90)

empno	ename	deptno	salary	roomno
E02	Morgan	D01	70	R03
E03	Robert	D01	80	R02
E05	Lincoln	D02	90	R11
E06	Taylor	D01	80	R12
E93	Matsuda	D91	90	R91

emp2 (salary \geq 90)

empno	ename	deptno	salary	roomno
E01	Smith	D01	100	R01
E04	Washington	D02	120	R10
E05	Lincoln	D02	90	R11
E91	Suzuki	D91	140	R91
E92	Tanaka	D91	130	R91
E93	Matsuda	D91	90	R91

```
select * from emp1
```

```
union select * from emp2
```

empno	ename	deptno	salary	roomno
E01	Smith	D01	100	R01
E02	Morgan	D01	70	R03
E03	Robert	D01	80	R02
E04	Washington	D02	120	R10
E05	Lincoln	D02	90	R11
E06	Taylor	D01	80	R12
E91	Suzuki	D91	140	R91
E92	Tanaka	D91	130	R91
E93	Matsuda	D91	90	R91

```
select * from emp1
```

```
union all select * from emp2
```

empno	ename	deptno	salary	roomno
E02	Morgan	D01	70	R03
E03	Robert	D01	80	R02
E05	Lincoln	D02	90	R11
E06	Taylor	D01	80	R12
E93	Matsuda	D91	90	R91
E01	Smith	D01	100	R01
E04	Washington	D02	120	R10
E05	Lincoln	D02	90	R11
E91	Suzuki	D91	140	R91
E92	Tanaka	D91	130	R91
E93	Matsuda	D91	90	R91

select * from emp1

intersect select * from emp2

empno	ename	deptno	salary	roomno
E05	Lincoln	D02	90	R11
E93	Matsuda	D91	90	R91

select * from emp1

except select * from emp2

empno	ename	deptno	salary	roomno
E02	Morgan	D01	70	R03
E03	Robert	D01	80	R02
E06	Taylor	D01	80	R12

商演算をSQLでどう表現するか？

- SQLでは商演算に相当する演算子がないため、直積、差集合、射影の組合せで表現する
- リレーショナル代数式 $\pi_X(\pi_X(R) \times S)$ をSQLでどう書くか？

$R \times S$: `select R.*, S.* from R, S`

$\pi_X(R) \times S$: `select distinct R.X, S.* from R, S`

$\pi_X(\pi_X(R) \times S)$:

`select distinct a.X from`

`(select distinct R.X, S.* from R, S) a`

(注) PostgreSQLではfromの後に副問合せが来るときには**別名**が必要 (Oracleでは不要)

商演算の表現

$$R \div S = \pi_X(R) - \pi_X\left(\underbrace{\left(\underbrace{\pi_X(R)}_{(a)} \times S\right)}_{(b)} - R\right)$$

(c)

(d)

R	X	Y
	a	1
	a	2
	b	1

S	Y
	1
	2

(a)	X	Y
	a	1
	a	2
	b	1
	b	2

(b)	X	Y
	b	2

(c)	X
	b

(d)	X
	a

$$R \div S = \pi_X(R) - \pi_X(\underbrace{(\underbrace{\underbrace{(\pi_X(R) \times S)}_{(a)}}_{(b)}}_{(c)}) - R)_{(d)}$$

select distinct a.X from R a
 except select distinct e.X from
 (select distinct b.X, c.Y from R b, S c
 except select d.* from R d) e

R	
X	Y
a	1
a	2
b	1

S
Y
1
2

(a)	
X	Y
a	1
a	2
b	1
b	2

(b)	
X	Y
b	2

(c)
X
b

(d)
X
a