Leckék Oracle gyakorlatra
./sql/lecke04_csoportok.pdf

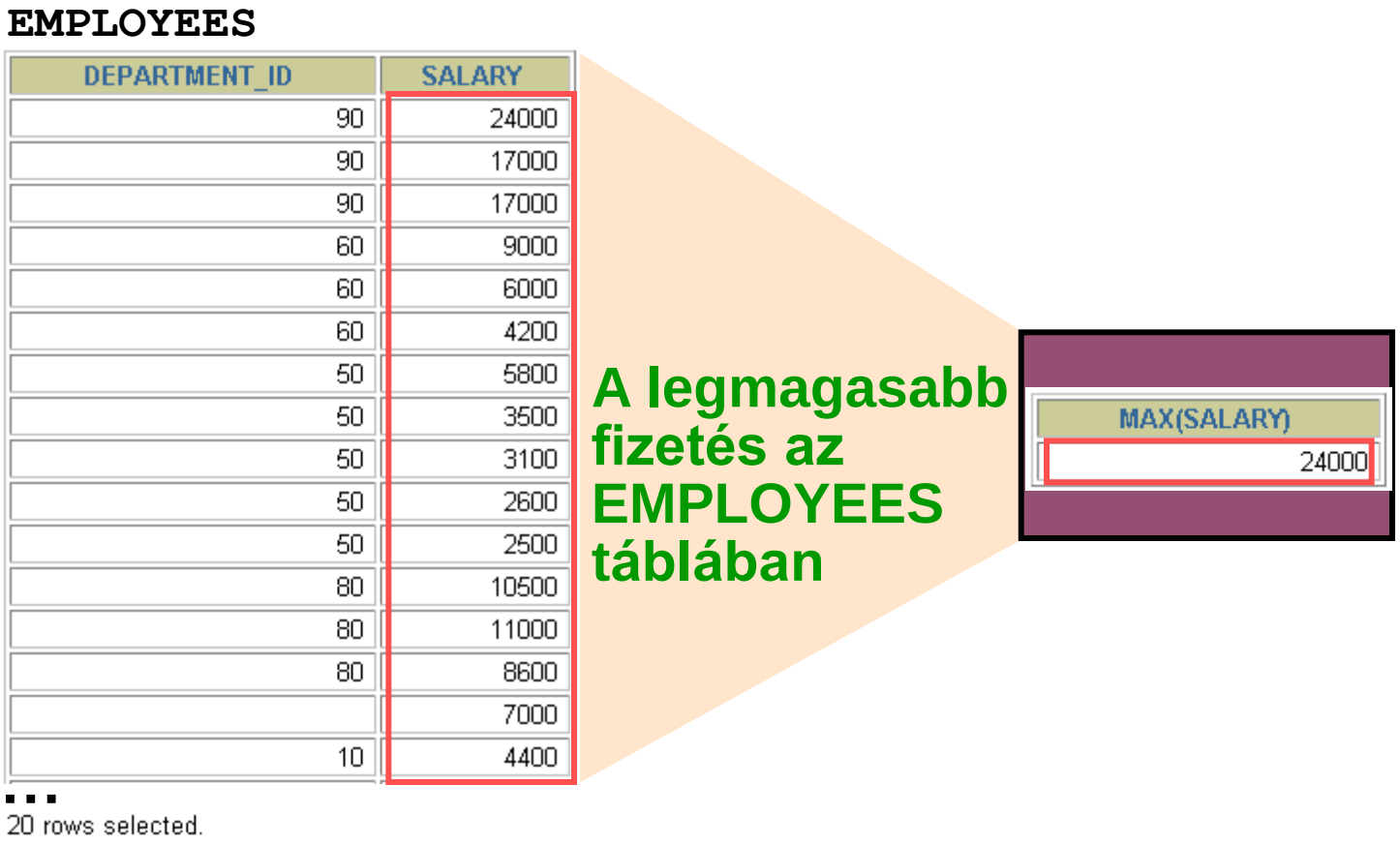
Csoportosított adatok megjelenítése összesítő függvények használatával

Célkitűzés

- A használható összesítő függvények azonosítása
- Az összesítő függvények használatának leírása
- Adatok csoportosítása a GROUP BY résszel
- A csoportosított sorok szűrése a HAVING résszel

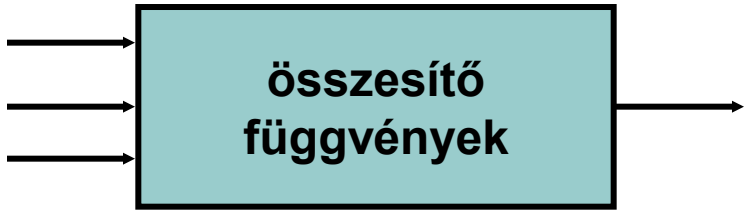
Az összesítő függvények azonosítása

- az összesítő függvény csoportosított sorok halmazain működik, és egyetlen eredményt ad vissza csoportonként.



Az aggregáló függvények

- AVG
- COUNT
- MAX
- MIN
- STDDEV
- SUM
- VARIANCE
- … statisztikai aggregáló függvények



Az aggregáló függvények (folyt.)

Függvény	Leírása
AVG([DISTINCT|ALL]*n*)	**n átlagértéke (nullértékeket kihagyva)**
COUNT({*|[DISTINCT|ALL]*expr*})	**Azon sorok száma, amelyekre *expr* kiértékelése nem null (DE: * esetén az összes sorok száma, beleértve az ismétlődőket és a null értéket tartalmazókat is)**
MAX([DISTINCT|ALL]*expr*)	***Expr* legnagyobb értéke (nullértékeket kihagyva)**
MIN([DISTINCT|ALL]*expr*)	***Expr* legkisebb értéke (nullértékeket kihagyva)**
STDDEV([DISTINCT|ALL]*x*)	***n* szórása (nullértékeket kihagyva)**
SUM([DISTINCT|ALL]*n*)	***n* értékeinek összege (nullértékeket kihagyva)**
VARIANCE([DISTINCT|ALL]*x*)	***n* szórásnégyzete (nullértékeket kihagyva)**

Az aggregáló függvények használata

```
SELECT      [column,] oszlop_függvény(column), ...
FROM        table
[WHERE      condition]
[GROUP BY   column]
[ORDER BY   column];
```

Az összesítő függvények a nullértéket tartalmazó sorokat kihagyják, a nullérték helyettesítésére használható az NVL vagy a COALESCE függvény.

AVG és SUM összesítő függvény

- Az AVG és SUM csak numerikus adatokra használható (a VARIANCE és STDDEV szintén).

```
SELECT  AVG(salary), MAX(salary),
        MIN(salary), SUM(salary)
FROM    employees
WHERE   job_id LIKE '%REP%';
```

AVG(SALARY)	MAX(SALARY)	MIN(SALARY)	SUM(SALARY)
8150	11000	6000	32600

MIN és MAX összesítő függvény

- A MIN és MAX numerikus, karakteres és dátum típusú adatokra használható (LOB és LONG típusokra nem).

```
SELECT MIN(hire_date), MAX(hire_date)
FROM       employees;
```

MIN(HIRE_	MAX(HIRE_
17-JUN-87	29-JAN-00

COUNT összesítő függvény

- COUNT(*) visszaadja a sorok számát a táblában:

1

```
SELECT COUNT(*)
FROM   employees
WHERE  department_id = 50;
```

COUNT(*)
5

- COUNT(expr) azoknak a soroknak a számát adja vissza, amelyekben expr nem nullérték:

2

```
SELECT COUNT(commission_pct)
FROM   employees
WHERE  department_id = 80;
```

COUNT(COMMISSION_PCT)
3

A DISTINCT kulcsszó használata

- COUNT(DISTINCT expr) azoknak a soroknak a számát adja vissza, amelyekben expr értéke különböző és nem nullérték
- Pl. a különböző (nem null) osztályazonosítók száma az EMPLOYEES táblában:

```
SELECT COUNT(DISTINCT department_id)
FROM   employees;
```

COUNT(DISTINCTDEPARTMENT_ID)
7

Összesítő függvények és a nullértékek

- Az összesítő függvények általában nem veszik figyelembe a nullértéket tartalmazó sorokat:

1

```
SELECT AVG(commission_pct)
FROM   employees;
```

AVG(COMMISSION_PCT)
.2125

- A NVL függvénnyel kikényszeríthető a nullértéket tartalmazó sorok figyelembe vétele:

2

```
SELECT AVG(NVL(commission_pct, 0))
FROM   employees;
```

AVG(NVL(COMMISSION_PCT,0))
.0425

Adatcsoportok létrehozása

EMPLOYEES

DEPARTMENT_ID	SALARY
10	4400
20	13000
20	6000
50	5800
50	3500
50	3100
50	2500
50	2600
60	9000
60	6000
60	4200
80	10500
80	8600
80	11000
90	24000
90	17000

...

20 rows selected.

4400

9500

3500

6400

10033

Az EMPLOYEES tábla osztályai és azokon az átlagfizetések

DEPARTMENT_ID	AVG(SALARY)
10	4400
20	9500
50	3500
60	6400
80	10033.3333
90	19333.3333
110	10150
	7000

Adatcsoportok létrehozása: a GROUP BY rész szintaxisa

```
SELECT     column, group_function(column)
FROM       table
[WHERE     condition]
[GROUP BY  group_by_expression]
[ORDER BY  column];
```

- Egy tábla sorai csoportosíthatóak a GROUP BY rész használatával.
- A GROUP BY részben oszlop másodnevek nem szerepelhetnek.

A GROUP BY rész használata

- A SELECT lista minden olyan oszlopnevének, amely nem összesítő függvényekben fordul elő, szerepelnie kell a GROUP BY részben.

```
SELECT    department_id, AVG(salary)
FROM      employees
GROUP BY department_id ;
```

DEPARTMENT_ID	AVG(SALARY)
10	4400
20	9500
50	3500
60	6400
80	10033.3333
90	19333.3333
110	10150
	7000

8 rows selected.

A GROUP BY rész használata (folyt.)

- A GROUP BY oszlopneveknek nem kötelező szerepelni a SELECT listában.

```
SELECT    AVG(salary)
FROM      employees
GROUP BY department_id ;
```

AVG(SALARY)
4400
9500
3500
6400
10033.3333
19333.3333
10150
7000

összesítő függvény szerepelhet az ORDER BY részben is.

Csoportosítás több oszlopnév alapján

EMPLOYEES

DEPARTMENT_ID	JOB_ID	SALARY
90	AD_PRES	24000
90	AD_VP	17000
90	AD_VP	17000
60	IT_PROG	9000
60	IT_PROG	6000
60	IT_PROG	4200
50	ST_MAN	5800
50	ST_CLERK	3500
50	ST_CLERK	3100
50	ST_CLERK	2600
50	ST_CLERK	2500
80	SA_MAN	10500
80	SA_REP	11000
80	SA_REP	8600

...

20	MK_REP	6000
110	AC_MGR	12000
110	AC_ACCOUNT	8300

20 rows selected.

Az EMPLOYEES tábla osztályain az egyes beosztások átlagfizetései

DEPARTMENT_ID	JOB_ID	SUM(SALARY)
10	AD_ASST	4400
20	MK_MAN	13000
20	MK_REP	6000
50	ST_CLERK	11700
50	ST_MAN	5800
60	IT_PROG	19200
80	SA_MAN	10500
80	SA_REP	19600
90	AD_PRES	24000
90	AD_VP	34000
110	AC_ACCOUNT	8300
110	AC_MGR	12000
	SA_REP	7000

13 rows selected.

A GROUP BY használata több oszlopnév esetén

```
SELECT    department_id dept_id, job_id, SUM(salary)
FROM      employees
GROUP BY department_id, job_id ;
```

DEPT_ID	JOB_ID	SUM(SALARY)
10	AD_ASST	4400
20	MK_MAN	13000
20	MK_REP	6000
50	ST_CLERK	11700
50	ST_MAN	5800
60	IT_PROG	19200
80	SA_MAN	10500
80	SA_REP	19600
90	AD_PRES	24000
90	AD_VP	34000
110	AC_ACCOUNT	8300
110	AC_MGR	12000
	SA_REP	7000

13 rows selected.

Aggregátort tartalmazó szabálytalan lekérdezés

- Bármely oszlopnévnek vagy kifejezésnek a SELECT listában, ha az nem aggregáló függvény, szerepelnie kell a GROUP BY részben:

```
SELECT department_id, COUNT(last_name)
FROM   employees;
```

```
SELECT department_id, COUNT(last_name)
       *
ERROR at line 1:
not a single-group group function
```

Az oszlopnév nem szerepel a `GROUP BY` részben!

Aggregátort tartalmazó szabálytalan lekérdezés

- A csoportok korlátozására a WHERE feltétel nem használható.
- A HAVING rész szolgál a csoportok korlátozására.
- összesítő függvény a WHERE feltételben nem szerepelhet.

```
SELECT   department_id, AVG(salary)
FROM     employees
WHERE    AVG(salary) > 8000
GROUP BY department_id;
```

```
WHERE  AVG(salary) > 8000
       *
ERROR at line 3:
group function is not allowed here
```

A csoportok korlátozására a `WHERE` feltétel nem használható!

Csoportok korlátozása

EMPLOYEES

DEPARTMENT_ID	SALARY
90	24000
90	17000
90	17000
60	9000
60	6000
60	4200
50	5800
50	3500
50	3100
50	2600
50	2500
80	10500
80	11000
80	8600
...	
20	6000
110	12000
110	8300

20 rows selected.

A legmagasabb fizetés osztályonként, ha az nagyobb mint $10,000

DEPARTMENT_ID	MAX(SALARY)
20	13000
80	11000
90	24000
110	12000

Csoportok korlátozása: HAVING

- A HAVING rész használata esetén az adatbázis szerver az alábbiak szerint korlátozza a csoportokat:
  1. Csoportosítja a sorokat.
  2. Alkalmazza Az összesítő függvényeket a csoportokra.
  3. A HAVING résznek megfelelő csoportokat megjeleníti.

```
SELECT     column, group_function
FROM       table
[WHERE     condition]
[GROUP BY  group_by_expression]
[HAVING    group_condition]
[ORDER BY  column];
```

A HAVING rész használata

```
SELECT    department_id, MAX(salary)
FROM      employees
GROUP BY  department_id
HAVING    MAX(salary)>10000 ;
```

DEPARTMENT_ID	MAX(SALARY)
20	13000
80	11000
90	24000
110	12000

A HAVING rész használata (folyt.)

```
SELECT   job_id, SUM(salary) PAYROLL
FROM     employees
WHERE    job_id NOT LIKE '%REP%'
GROUP BY job_id
HAVING   SUM(salary) > 13000
ORDER BY SUM(salary);
```

JOB_ID	PAYROLL
IT_PROG	19200
AD_PRES	24000
AD_VP	34000

összesítő függvények egymásba ágyazása

- Az osztályonkénti legmagasabb átlagfizetés megjelenítése:

```
SELECT    MAX(AVG(salary))
FROM      employees
GROUP BY department_id;
```

MAX(AVG(SALARY))
19333.3333

Az összesítő függvények csak kétszeres mélységig ágyazhatóak egymásba!

Összefoglalás

- Ebben a részben megtanultuk:
  - a COUNT, MAX, MIN, és AVG összesítő függvények használatát,
  - hogyan írjunk GROUP BY részt tartalmazó lekérdezéseket,
  - hogyan írjunk HAVING részt tartalmazó lekérdezéseket.

```
SELECT     column, group_function
FROM       table
[WHERE     condition]
[GROUP BY  group_by_expression]
[HAVING    group_condition]
[ORDER BY  column];
```