Universitatea Constantin Brâncuşi din Târgu Jiu
Facultatea: Inginerie
Program de conversie profesională a cadrelor didactice din învățământul preuniversitar: Informatică, Tehnologia Informației și a Comunicațiilor

Baze de date

Limbajul SQL

Teams: TIC - Baze de date-2022/2023

Adrian Runceanu

Dr. Adrian Runceanu

copyright@www.adrian.runceanu.ro

ORACLE

THE INFORMATION COMPANY

Curs 4

Limbajul SQL

copyright@www.adrian.runceanu.ro

Limbajul SQL

Cereri SELECT pe o tabela

4.1. Funcţii

4.2. Funcţii referitoare la o singură înregistrare

4.3. Funcţii referitoare la mai multe înregistrări

- **4.3.1. Clauza GROUP BY**
- **4.3.2. Excluderea grupurilor (clauza HAVING)**
- **4.3.3. Imbricarea funcţiilor de grup**

copyright@www.adrian.runceanu.ro

Tabele EMP si DEPT

Pentru exemplele din cursuri vom folosi tabelele ***EMP*** si ***DEPT***.

EMP

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST | Sample Queries

Add Column | Modify Column | Rename Column | Drop Column | Rename | Copy | Drop | Truncate | Create Lookup Table | Create App

Column Name	Data Type	Nullable	Default	Primary Key
EMPNO	NUMBER(4,0)	No	-	1
ENAME	VARCHAR2(50)	Yes	-	-
JOB	VARCHAR2(50)	Yes	-	-
MGR	NUMBER(4,0)	Yes	-	-
HIREDATE	DATE	Yes	-	-
SAL	NUMBER(7,2)	Yes	-	-
COMM	NUMBER(7,2)	Yes	-	-
DEPTNO	NUMBER(2,0)	Yes	-	-

copyright@www.adrian.runceanu.ro

Tabele EMP si DEPT

Pentru exemplele din cursuri vom folosi tabelele ***EMP*** si ***DEPT***.

EMP

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST | Sample Queries

Query | Count Rows | Insert Row | Load Data

EDIT	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
	7839	KING	PRESIDENT	-	11/17/1981	34475.65	-	10
	7698	BLAKE	MANAGER	7839	05/01/1981	19651.14	-	30
	7782	CLARK	MANAGER	7839	06/09/1981	16893.06	-	10
	7566	JONES	MANAGER	7839	04/02/1981	20513.04	-	20
	7788	SCOTT	ANALYST	7566	12/09/1982	20685.39	-	20
	7902	SMITH	ANALYST	7566	12/03/1981	20685.39	-	20
	7369	SMITH	CLERK	7902	12/17/1980	5516.12	-	20
	7499	ALLEN	SALESMAN	7698	02/20/1981	11032.2	300	30
	7521	WARD	SALESMAN	7698	02/22/1981	8618.96	500	30
	7654	SMITH	SALESMAN	7698	09/28/1981	8618.96	1400	30
	7844	TURNER	SALESMAN	7698	09/08/1981	10342.72	0	30
	7876	ADAMS	CLERK	7788	01/12/1983	7584.64	-	20
	7900	JAMES	CLERK	7698	12/03/1981	6550.4	-	30
	7934	MILLER	CLERK	7782	01/23/1982	8963.67	-	10

copyright@www.adrian.runceanu.ro

Tabele EMP si DEPT

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST | Sample Queries

Add Column | Modify Column | Rename Column | Drop Column | Rename | Copy | Drop | Truncate | Create Lookup Table | Create App

Column Name	Data Type	Nullable	Default	Primary Key
DEPTNO	NUMBER(2,0)	No	-	1
DNAME	VARCHAR2(50)	Yes	-	-
LOC	VARCHAR2(50)	Yes	-	-

DEPT

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST | Sample Queries

Query | Count Rows | Insert Row | Load Data

EDIT	DEPTNO	DNAME	LOC
	10	ACCOUNTING	NEW YORK
	20	RESEARCH	DALLAS
	30	SALES	CHICAGO
	40	OPERATIONS	BOSTON
	15	Departament IT	123

copyright@www.adrian.runceanu.ro

Funcţii

Funcţiile sunt o caracteristică importantă a SQL si sunt utilizate pentru:

- ✓ a realiza calcule asupra datelor
- ✓ a modifica date
- ✓ a manipula grupuri de înregistrări
- ✓ a schimba formatul datelor
- ✓ sau pentru a converti diferite tipuri de date

copyright@www.adrian.runceanu.ro

Funcţii

Funcţiile se clasifică în două tipuri:

1. Funcţii referitoare la o singură înregistrare (single-row functions)

2. Funcţii referitoare la mai multe înregistrări (multiple-row functions)

copyright@www.adrian.runceanu.ro

Funcţii

1. Funcţii referitoare la o singură înregistrare (single-row functions):

1. **funcţii caracter**
2. **funcţii numerice**
3. **funcţii pentru data calendaristică si oră**
4. **funcţii de conversie**
5. **funcţii diverse**

copyright@www.adrian.runceanu.ro

Funcţii

2. Funcţii referitoare la mai multe înregistrări (multiple-row functions):

- funcţii totalizatoare sau funcţii de grup

copyright@www.adrian.runceanu.ro

Funcţii

Diferenţa dintre cele două tipuri de funcţii este numărul de înregistrări pe care acţionează:

- *Funcţiile referitoare la o singură înregistrare returnează un singur rezultat pentru fiecare rând al tabelei*,
- pe când *funcţiile referitoare la mai multe înregistrări returnează un singur rezultat pentru fiecare grup de înregistrări din tabela*.

copyright@www.adrian.runceanu.ro

Funcţii

O observaţie importanta este faptul că dacă se apelează o funcţie **SQL** ce are un argument (parametru) egal cu valoarea Null, atunci în mod automat rezultatul va avea valoarea Null.

Singurele funcţii care nu respectă această regulă sunt:

- **CONCAT**
- **DECODE**
- **DUMP**
- **NVL**
- **REPLACE**

copyright@www.adrian.runceanu.ro

Limbajul SQL

Cereri SELECT pe o tabela

4.1. Funcţii

4.2. Funcţii referitoare la o singură înregistrare

4.3. Funcţii referitoare la mai multe înregistrări

- 4.3.1. Clauza GROUP BY
- 4.3.2. Excluderea grupurilor (clauza HAVING)
- 4.3.3. Imbricarea funcţiilor de grup

copyright@www.adrian.runceanu.ro

Funcţii referitoare la o singură înregistrare

Sunt funcţii utilizate pentru manipularea datelor individuale.

Ele pot avea unul sau mai multe argumente şi returnează o valoare pentru fiecare rând rezultat în urma interogării.

copyright@www.adrian.runceanu.ro

Funcţii referitoare la o singură înregistrare

Sunt mai multe tipuri de funcţii pe un singur rând.

Sintaxa:

function_name [(arg1,arg2,...)]

În sintaxa generală *function_name* este numele funcţiei şi *arg1,arg2* sunt argumentele funcţiei care pot fi date de numele unei coloane sau de o expresie.

copyright@www.adrian.runceanu.ro

Funcţii referitoare la o singură înregistrare

Funcţiile pe un singur rând cuprind următoarele tipuri de funcţii:

1. funcţii de tip caracter
2. funcţii de tip numeric
3. funcţii de tip data
4. funcţii de conversie
5. funcţii generale: NVL, NVL2, NULLIF, COALESCE, CASE, DECODE

copyright@www.adrian.runceanu.ro

Funcţii referitoare la o singură înregistrare

1. Funcţii de tip caracter

Aceste funcţii au ca argumente date de tip caracter şi returnează date de tip VARCHAR2, CHAR sau NUMBER.

Funcţii referitoare la o singură înregistrare

Cele mai importante funcţii caracter sunt:

Functie	Descriere
LOWER(column|expression)	converteste alfa caracterele din caractere mari in caractere mici
UPPER(column|expression)	converteste alfa caracterele din caractere mici in caractere mari
INITCAP(column|expression)	converteste prima litera a fiecarui cuvant in caractere mari si restul cuvantului in caractere mici
CONCAT(column1|expression1, column2|expression2)	functia este echivalentul operatorului de concantenare (||)
SUBSTR(column|expression, m [, n])	returneaza un sir de *n* caractere incepand cu caracterul aflat pe pozitia *m*
LENGTH(column|expression)	returneaza numarul de caractere dintr-o expresie
INSTR(column|expression, 'string', [m], [n])	returneaza pozitia unui anumit sir, optional se poate incepe cautarea cu pozitia *m* sau cu a *n*-a aparitie a sirului. *m* si *n* sunt prin definitie 1
REPLACE(text, search_string, replacement_ string)	cauta un anumit text intr-un sir de caractere si daca il gaseste il inlocuieste

copyright@www.adrian.runceanu.ro

Funcţii referitoare la o singură înregistrare

Exemplu de utilizare a funcţiei LENGTH:

Column Name	Data Type	Nullable	Default	Primary Key
EMPNO	NUMBER(4,0)	No		1
ENAME	VARCHAR2(50)	Yes		
JOB	VARCHAR2(50)	Yes		
MGR	NUMBER(4,0)	Yes		
HIREDATE	DATE	Yes		
SAL	NUMBER(7,2)	Yes		
COMM	NUMBER(7,2)	Yes		
DEPTNO	NUMBER(2,0)	Yes		

```
SELECT LENGTH(ename)
FROM EMP;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		11/17/1981	5000		10
7698	BLAKE	MANAGER	7839	5/1/1981	2850		30
7782	CLARK	MANAGER	7839	6/9/1981	2450		10
7566	JONES	MANAGER	7839	4/2/1981	2975		20
7788	SCOTT	ANALYST	7566	12/9/1982	3000		20
7902	FORD	ANALYST	7566	12/3/1981	3000		20
7369	SMITH	CLERK	7902	12/17/1980	800		20
7499	ALLEN	SALESMAN	7698	2/20/1981	1600	300	30
7521	WARD	SALESMAN	7698	2/22/1981	1250	500	30
7654	MARTIN	SALESMAN	7698	9/28/1981	1250	1400	30
7844	TURNER	SALESMAN	7698	9/8/1981	1500	0	30
7876	ADAMS	CLERK	7788	1/12/1983	1100		20
7900	JAMES	CLERK	7698	12/3/1981	950		30
7934	MILLER	CLERK	7782	1/23/1982	1300		10

copyright@www.adrian.runceanu.ro

Funcţii referitoare la o singură înregistrare

Exemplu de utilizare a funcţiei LENGTH – rezultatul obtinut:

```
1    SELECT LENGTH(ename)
2    FROM EMP;
```

Results | Explain | Describe | Saved SQL | History

LENGTH(ENAME)
4
5
5
5
5
4
5
5
4
6
6
5
5
6

14 rows returned in 0.03 seconds Download

copyright@www.adrian.runceanu.ro

Funcţii referitoare la o singură înregistrare

Exemplu:

```
SELECT 'Numele functiei pentru '||UPPER(ename)||'este
   '||LOWER(job) AS "DETALII ANGAJAT"
FROM EMP;
```

Column Name	Data Type	Nullable	Default	Primary Key
EMPNO	NUMBER(4,0)	No		1
ENAME	VARCHAR2(50)	Yes		
JOB	VARCHAR2(50)	Yes		
MGR	NUMBER(4,0)	Yes		
HIREDATE	DATE	Yes		
SAL	NUMBER(7,2)	Yes		
COMM	NUMBER(7,2)	Yes		
DEPTNO	NUMBER(2,0)	Yes		

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		11/17/1981	5000		10
7698	BLAKE	MANAGER	7839	5/1/1981	2850		30
7782	CLARK	MANAGER	7839	6/9/1981	2450		10
7566	JONES	MANAGER	7839	4/2/1981	2975		20
7788	SCOTT	ANALYST	7566	12/9/1982	3000		20
7902	FORD	ANALYST	7566	12/3/1981	3000		20
7369	SMITH	CLERK	7902	12/17/1980	800		20
7499	ALLEN	SALESMAN	7698	2/20/1981	1600	300	30
7521	WARD	SALESMAN	7698	2/22/1981	1250	500	30
7654	MARTIN	SALESMAN	7698	9/28/1981	1250	1400	30
7844	TURNER	SALESMAN	7698	9/8/1981	1500	0	30
7876	ADAMS	CLERK	7788	1/12/1983	1100		20
7900	JAMES	CLERK	7698	12/3/1981	950		30
7934	MILLER	CLERK	7782	1/23/1982	1300		10

copyright@www.adrian.runceanu.ro

Funcţii referitoare la o singură înregistrare

Rezultat obtinut:

```
1  SELECT 'Numele functiei pentru '||UPPER(ename)||'  este '||LOWER(job) AS "DETALII ANGAJAT"
2  FROM EMP;
```

Results | Explain | Describe | Saved SQL | History

DETALII ANGAJAT
Numele functiei pentru KING este president
Numele functiei pentru BLAKE este manager
Numele functiei pentru CLARK este manager
Numele functiei pentru JONES este manager
Numele functiei pentru SCOTT este analyst
Numele functiei pentru FORD este analyst
Numele functiei pentru SMITH este clerk
Numele functiei pentru ALLEN este salesman
Numele functiei pentru WARD este salesman
Numele functiei pentru MARTIN este salesman
Numele functiei pentru TURNER este salesman
Numele functiei pentru ADAMS este clerk
Numele functiei pentru JAMES este clerk
Numele functiei pentru MILLER este clerk

14 rows returned in 0.00 seconds Download

copyright@www.adrian.runceanu.ro

Funcţii referitoare la o singură înregistrare

Exemplu:

```
SELECT empno, UPPER(ename), job, deptno
FROM EMP
WHERE ename = 'MARTIN';
```

Column Name	Data Type	Nullable	Default	Primary Key
EMPNO	NUMBER(4,0)	No		1
ENAME	VARCHAR2(50)	Yes		
JOB	VARCHAR2(50)	Yes		
MGR	NUMBER(4,0)	Yes		
HIREDATE	DATE	Yes		
SAL	NUMBER(7,2)	Yes		
COMM	NUMBER(7,2)	Yes		
DEPTNO	NUMBER(2,0)	Yes		

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		11/17/1981	5000		10
7698	BLAKE	MANAGER	7839	5/1/1981	2850		30
7782	CLARK	MANAGER	7839	6/9/1981	2450		10
7566	JONES	MANAGER	7839	4/2/1981	2975		20
7788	SCOTT	ANALYST	7566	12/9/1982	3000		20
7902	FORD	ANALYST	7566	12/3/1981	3000		20
7369	SMITH	CLERK	7902	12/17/1980	800		20
7499	ALLEN	SALESMAN	7698	2/20/1981	1600	300	30
7521	WARD	SALESMAN	7698	2/22/1981	1250	500	30
7654	MARTIN	SALESMAN	7698	9/28/1981	1250	1400	30
7844	TURNER	SALESMAN	7698	9/8/1981	1500	0	30
7876	ADAMS	CLERK	7788	1/12/1983	1100		20
7900	JAMES	CLERK	7698	12/3/1981	950		30
7934	MILLER	CLERK	7782	1/23/1982	1300		10

copyright@www.adrian.runceanu.ro

Funcţii referitoare la o singură înregistrare

Rezultat obtinut:

```
1    SELECT empno, UPPER(ename), job, deptno
2    FROM EMP
3    WHERE ename = 'MARTIN';
```

Results | Explain | Describe | Saved SQL | History

EMPNO	UPPER(ENAME)	JOB	DEPTNO
7654	MARTIN	SALESMAN	30

1 rows returned in 0.01 seconds Download

copyright@www.adrian.runceanu.ro

Funcţii referitoare la o singură înregistrare

Clauza **WHERE** a acestei cereri SQL compară numele din tabela Angajaţi cu 'Smith'.

Pentru comparaţie numele sunt convertite în litere mici şi din această cauză se obţine un rezultat.

Exemplu:

```
SELECT empno, UPPER(ename), job, deptno
FROM EMP
WHERE INITCAP(ename) = 'Smith';
```

copyright@www.adrian.runceanu.ro

Funcţii referitoare la o singură înregistrare

Rezultatul obtinut:

```
1    SELECT empno, UPPER(ename), job, deptno
2    FROM EMP
3    WHERE INITCAP(ename) = 'Smith';
```

Results | Explain | Describe | Saved SQL | History

EMPNO	UPPER(ENAME)	JOB	DEPTNO
7369	SMITH	CLERK	20

1 rows returned in 0.00 seconds Download

copyright@www.adrian.runceanu.ro

Funcţii referitoare la o singură înregistrare

Exemplu:

Pentru afişarea numelui cu majuscule de foloseşte funcţia **UPPER**.

```
SELECT empno, CONCAT(ename, job), ename,
   UPPER(ename)
FROM EMP;
```

copyright@www.adrian.runceanu.ro

Funcţii referitoare la o singură înregistrare

Rezultatul obtinut:

```
1    SELECT empno, CONCAT(ename, job), ename, UPPER(ename)
2    FROM EMP;
```

Results | Explain | Describe | Saved SQL | History

EMPNO	CONCAT(ENAME,JOB)	ENAME	UPPER(ENAME)
7839	KINGPRESIDENT	KING	KING
7698	BLAKEMANAGER	BLAKE	BLAKE
7782	CLARKMANAGER	CLARK	CLARK
7566	JONESMANAGER	JONES	JONES
7788	SCOTTANALYST	SCOTT	SCOTT
7902	FORDANALYST	FORD	FORD
7369	SMITHCLERK	SMITH	SMITH
7499	ALLENSALESMAN	ALLEN	ALLEN
7521	WARDSALESMAN	WARD	WARD
7654	MARTINSALESMAN	MARTIN	MARTIN
7844	TURNERSALESMAN	TURNER	TURNER
7876	ADAMSCLERK	ADAMS	ADAMS
7900	JAMESCLERK	JAMES	JAMES
7934	MILLERCLERK	MILLER	MILLER

14 rows returned in 0.01 seconds Download

copyright@www.adrian.runceanu.ro

Funcţii referitoare la o singură înregistrare

Spre deosebire de alte funcţii, funcţiile caracter pot fi imbricate până la orice adâncime.

Dacă funcţiile sunt imbricate, atunci ele sunt evaluate din interior spre exterior.

Pentru a determina, de exemplu, de câte ori apare caracterul 'A' în câmpul nume vom folosi interogarea:

copyright@www.adrian.runceanu.ro

Funcţii referitoare la o singură înregistrare

```
SELECT ename, LENGTH (ename) - LENGTH (TRANSLATE(ename, 'DA',
    'D'))
FROM EMP;
```

Column Name	Data Type	Nullable	Default	Primary Key
EMPNO	NUMBER(4,0)	No		1
ENAME	VARCHAR2(50)	Yes		
JOB	VARCHAR2(50)	Yes		
MGR	NUMBER(4,0)	Yes		
HIREDATE	DATE	Yes		
SAL	NUMBER(7,2)	Yes		
COMM	NUMBER(7,2)	Yes		
DEPTNO	NUMBER(2,0)	Yes		

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		11/17/1981	5000		10
7698	BLAKE	MANAGER	7839	5/1/1981	2850		30
7782	CLARK	MANAGER	7839	6/9/1981	2450		10
7566	JONES	MANAGER	7839	4/2/1981	2975		20
7788	SCOTT	ANALYST	7566	12/9/1982	3000		20
7902	FORD	ANALYST	7566	12/3/1981	3000		20
7369	SMITH	CLERK	7902	12/17/1980	800		20
7499	ALLEN	SALESMAN	7698	2/20/1981	1600	300	30
7521	WARD	SALESMAN	7698	2/22/1981	1250	500	30
7654	MARTIN	SALESMAN	7698	9/28/1981	1250	1400	30
7844	TURNER	SALESMAN	7698	9/8/1981	1500	0	30
7876	ADAMS	CLERK	7788	1/12/1983	1100		20
7900	JAMES	CLERK	7698	12/3/1981	950		30
7934	MILLER	CLERK	7782	1/23/1982	1300		10

copyright@www.adrian.runceanu.ro

Funcţii referitoare la o singură înregistrare

Rezultatul obtinut:

```
1   SELECT ename, LENGTH (ename) - LENGTH (TRANSLATE(ename,'DA','D'))
2   FROM EMP;
```

Results | Explain | Describe | Saved SQL | History

ENAME	LENGTH(ENAME)-LENGTH(TRANSLATE(ENAME,'DA','D'))
KING	0
BLAKE	1
CLARK	1
JONES	0
SCOTT	0
FORD	0
SMITH	0
ALLEN	1
WARD	1
MARTIN	1
TURNER	0
ADAMS	2
JAMES	1
MILLER	0

14 rows returned in 0.00 seconds Download

copyright@www.adrian.runceanu.ro

Funcţii referitoare la o singură înregistrare

Explicatii:

În exemplul anterior, funcţia TRANSLATE (nume, 'DA', 'D') va căuta în coloana “nume” primul caracter (caracterul 'D') din cel de-al doilea argument al funcţiei (şirul de caractere 'DA') şi îl va înlocui cu primul caracter (adică tot cu caracterul 'D') din cel de-al treilea argument al funcţiei (şirul de caractere 'D'), apoi va căuta cel de-al doilea caracter, adică caracterul 'A', şi îl va şterge din câmpul nume deoarece acesta nu are caracter corespondent în cel de-al treilea argument al funcţiei.

Am folosit acest artificiu deoarece şirul de caractere vid este echivalent cu valoarea Null, deci funcţia TRANSLATE (nume, 'A', '') ar fi înlocuit toate valorile câmpului “nume” cu valoarea Null.

copyright@www.adrian.runceanu.ro

Limbajul SQL

Cereri SELECT pe o tabela

4.1. Funcţii

4.2. Funcţii referitoare la o singură înregistrare

4.3. Funcţii referitoare la mai multe înregistrări

- **4.3.1. Clauza GROUP BY**
- **4.3.2. Excluderea grupurilor (clauza HAVING)**
- **4.3.3. Imbricarea funcţiilor de grup**

copyright@www.adrian.runceanu.ro

4.3. Funcții de grup

Funcţiile de grup sunt funcţii care operează pe un set de rânduri pentru a da un rezultat pe întreg setul.

Parametrii şi descrierea funcţiilor de grup.

Funcţiile de grup sunt:

1. **AVG**
2. **COUNT**
3. **MAX**
4. **MIN**
5. **STDDEV**
6. **SUM**
7. **VARIANCE**

copyright@www.adrian.runceanu.ro

4.3. Funcții de grup

Fiecare dintre aceste funcţii acceptă anumiţi parametri:

Functia	Descriere
AVG([DISTINCT|ALL]n)	Valoarea medie pentru grup, ignorand valorile nule
COUNT({*|[DISTINCT|ALL]expr})	Numarul de randuri unde expr evalueaza altceva in afara de null (folosind * sunt numarate toate randurile, incluzand duplicatele si pe cele cu valoare nula)
MAX([DISTINCT|ALL]expr)	Valoarea maxima a expr, ignorand valorile nule
MIN([DISTINCT|ALL]expr)	Valoarea minima a expr, ignorand valorile nule
STDDEV([DISTINCT|ALL]x)	Deviatia standard pentru grup, ignorand valorile nule
SUM([DISTINCT|ALL]x)	Suma valorilor pentru grup, ignorand valorile nule
VARIANCE([DISTINCT|ALL]x)	Variatia pentru grup, ignorand valorile nule

copyright@www.adrian.runceanu.ro

4.3. Funcții de grup

DISTINCT face ca funcţia să ignore valorile duplicat.

ALL face ca funcţia să afişeze şi valorile duplicat.

Valoarea implicită este **ALL**, deci nu este necesar să fie specificată.

Tipul de dată returnat de funcţia expr poate fi **CHAR**, **VARCHAR2**, **NUMBER** sau **DATE**.

Toate funcţiile de grup ignoră valorile nule.

Pentru a lua în considerare şi valorile nule se folosesc funcţiile **NVL**, **NVL2** sau **COALESCE**.

copyright@www.adrian.runceanu.ro

4.3. Funcții de grup

Sintaxa funcţiilor de grup:

```
SELECT [coloana,] functie_de_grup(coloana),
...
FROM tabel
[WHERE conditie]
[GROUP BY coloana]
[HAVING conditie_de_grupare]
[ORDER BY coloana];
```

Rezultatele sunt sortate implicit crescător. Pentru o ordonare descrescătoare se va folosi clauza **DESC** după **ORDER BY**.

copyright@www.adrian.runceanu.ro

4.3. Funcții de grup

Exemplul 1: Afişarea salariului mediu, maxim, minim şi suma tuturor salariilor angajaţilor cu funcţie SALESMAN.

SELECT **AVG**(sal), **MAX**(sal), **MIN**(sal), **SUM**(sal)
FROM EMP
WHERE job = 'SALESMAN';

```
1   SELECT AVG(sal), MAX(sal), MIN(sal), SUM(sal)
2   FROM EMP
3   WHERE job = 'SALESMAN';
```

Results | Explain | Describe | Saved SQL | History

AVG(SAL)	MAX(SAL)	MIN(SAL)	SUM(SAL)
1400	1600	1250	5600

1 rows returned in 0.01 seconds Download

copyright@www.adrian.runceanu.ro

4.3. Funcții de grup

Exemplul 2 - Datele la care s-au făcut prima şi ultima angajare.

SELECT **MIN**(hiredate), **MAX**(hiredate)
FROM EMP;

```
1   SELECT MIN(hiredate), MAX(hiredate)
2   FROM EMP;
```

Results | Explain | Describe | Saved SQL | History

MIN(HIREDATE)	MAX(HIREDATE)
12/17/1980	01/12/1983

1 rows returned in 0.00 seconds Download

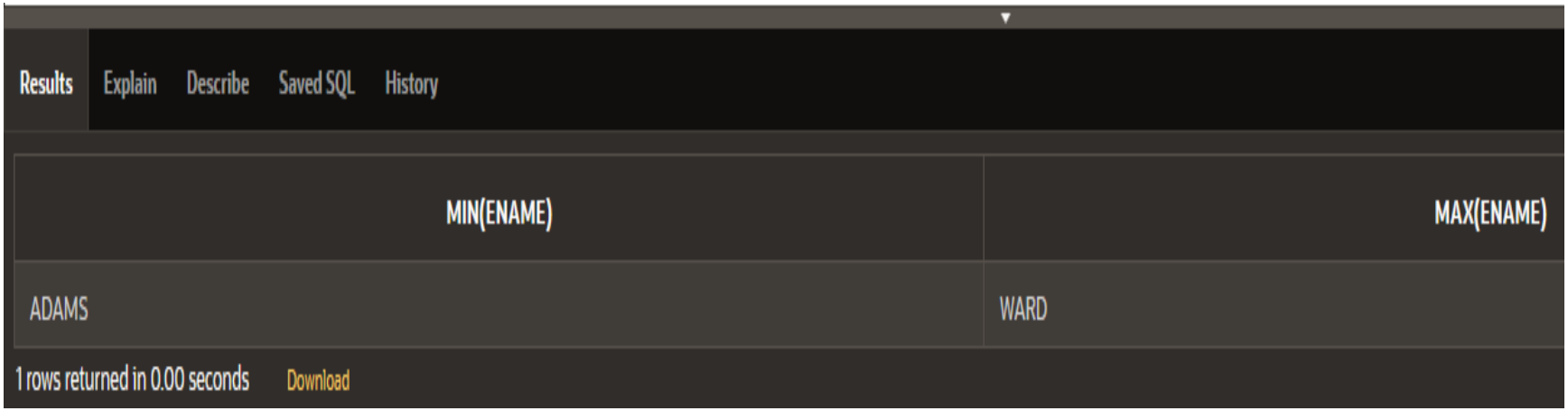
copyright@www.adrian.runceanu.ro

4.3. Funcții de grup

Exemplul 3 - Primul şi ultimul nume de angajat în ordine alfabetică:

SELECT **MIN**(ename), **MAX**(ename)
FROM EMP;

```
1    SELECT MIN(ename), MAX(ename)
2    FROM EMP;
```



copyright@www.adrian.runceanu.ro

4.3. Funcții de grup

Funcţia **COUNT**

Funcţia **COUNT** are 3 formate:

COUNT(*)
COUNT(expr)
COUNT(DISTINCT expr)

copyright@www.adrian.runceanu.ro

4.3. Funcții de grup

- **COUNT(*)** întoarce numărul de rânduri dintr-o tabela care satisface criteriul de selecţie, incluzând rândurile duplicat şi rândurile conţinând valori nule.

- Dacă clauza WHERE este introdusă, atunci **COUNT(*)** returnează numărul de rânduri care satisfac condiţia din clauza WHERE.

copyright@www.adrian.runceanu.ro

4.3. Funcții de grup

- În contrast, funcţia **COUNT(expr)** întoarce numărul de valori nenule din coloana specificată de expr.

- **COUNT(DISTINCT expr)** returnează numărul de valori distincte, nenule din coloana specificată de expr.

copyright@www.adrian.runceanu.ro

4.3. Funcții de grup

Exemplul 4

Numărul angajaţilor din departamentul cu id-ul 30.

SELECT COUNT(*)
FROM EMP
WHERE deptno = 30;

Column Name	Data Type	Nullable	Default	Primary Key
EMPNO	NUMBER(4,0)	No		1
ENAME	VARCHAR2(50)	Yes		
JOB	VARCHAR2(50)	Yes		
MGR	NUMBER(4,0)	Yes		
HIREDATE	DATE	Yes		
SAL	NUMBER(7,2)	Yes		
COMM	NUMBER(7,2)	Yes		
DEPTNO	NUMBER(2,0)	Yes		

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		11/17/1981	5000		10
7698	BLAKE	MANAGER	7839	5/1/1981	2850		30
7782	CLARK	MANAGER	7839	6/9/1981	2450		10
7566	JONES	MANAGER	7839	4/2/1981	2975		20
7788	SCOTT	ANALYST	7566	12/9/1982	3000		20
7902	FORD	ANALYST	7566	12/3/1981	3000		20
7369	SMITH	CLERK	7902	12/17/1980	800		20
7499	ALLEN	SALESMAN	7698	2/20/1981	1600	300	30
7521	WARD	SALESMAN	7698	2/22/1981	1250	500	30
7654	MARTIN	SALESMAN	7698	9/28/1981	1250	1400	30
7844	TURNER	SALESMAN	7698	9/8/1981	1500	0	30
7876	ADAMS	CLERK	7788	1/12/1983	1100		20
7900	JAMES	CLERK	7698	12/3/1981	950		30
7934	MILLER	CLERK	7782	1/23/1982	1300		10

copyright@www.adrian.runceanu.ro

4.3. Funcții de grup

Rezultatul obtinut - Numărul angajaţilor din departamentul cu id-ul 30.

```
1    SELECT COUNT(*)
2    FROM EMP
3    WHERE deptno = 30;
```

Results | Explain | Describe | Saved SQL | History

COUNT(*)
6

1 rows returned in 0.00 seconds Download

copyright@www.adrian.runceanu.ro

4.3. Funcții de grup

Exemplul 5

Numărul angajaţilor care iau comision din departamentul 30.

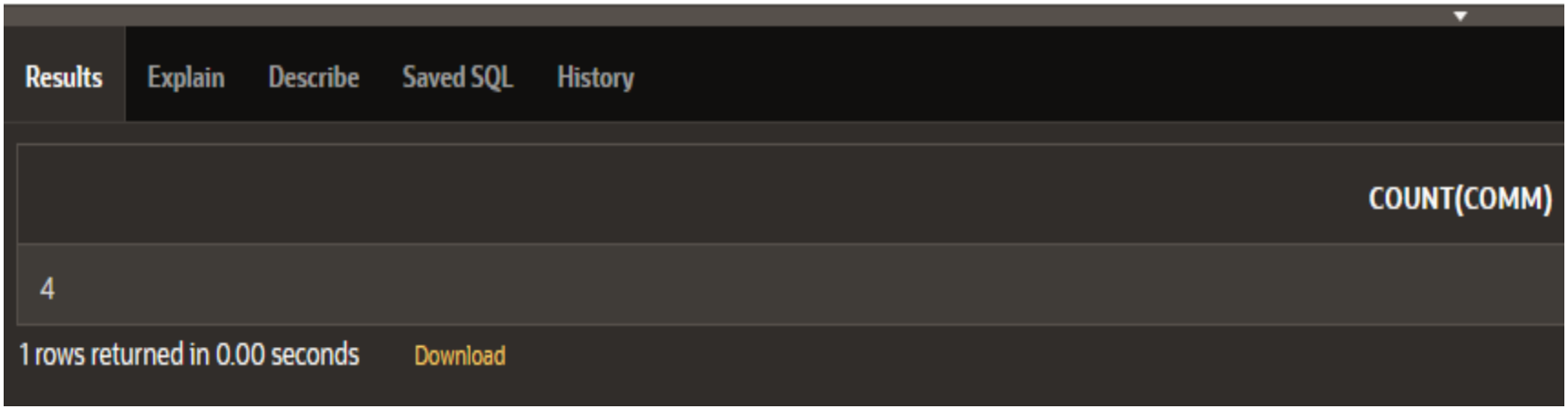
```
SELECT COUNT(comm)
FROM EMP
WHERE deptno = 30;
```

copyright@www.adrian.runceanu.ro

4.3. Funcții de grup

Rezultatul obtinut - Numărul angajaţilor care iau comision din departamentul 30:

```
1    SELECT COUNT(comm)
2    FROM EMP
3    WHERE deptno = 30;
```



copyright@www.adrian.runceanu.ro

4.3. Funcții de grup

Exemplul 6 - Numărul de departamente din firma (varianta incorectă şi varianta corectă).

SELECT COUNT(deptno), **COUNT**(**DISTINCT** deptno)
FROM EMP;

```
1  SELECT COUNT(deptno), COUNT(DISTINCT deptno)
2  FROM EMP;
3
```

Results | Explain | Describe | Saved SQL | History

COUNT(DEPTNO)	COUNT(DISTINCTDEPTNO)
14	3

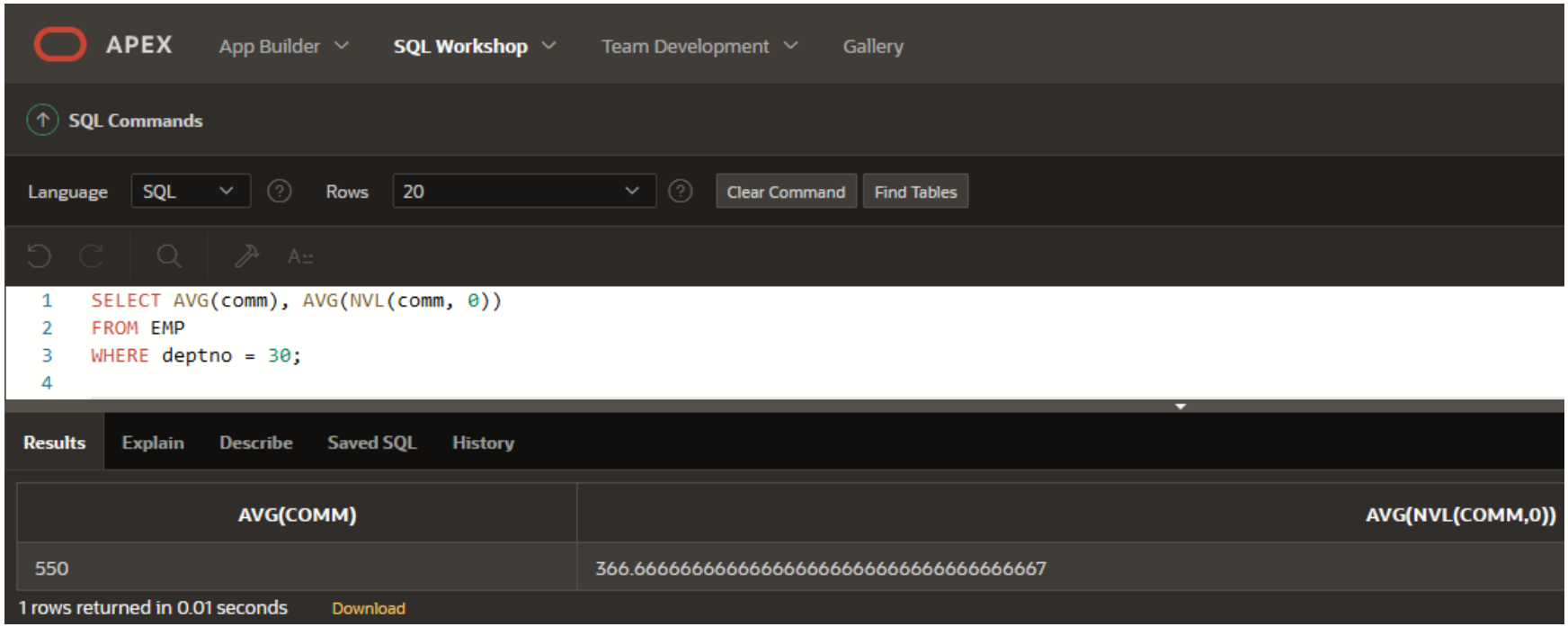
1 rows returned in 0.02 seconds Download

copyright@www.adrian.runceanu.ro

4.3. Funcții de grup

Exemplul 7 - Comisionul mediu în departamentul 30 (ignorând sau nu valorile nule):

```
SELECT AVG(comm), AVG(NVL(comm, 0))
FROM EMP
WHERE deptno = 30;
```



copyright@www.adrian.runceanu.ro

Limbajul SQL

Cereri SELECT pe o tabela

4.1. Funcţii

4.2. Funcţii referitoare la o singură înregistrare

4.3. Funcţii referitoare la mai multe înregistrări

- **4.3.1. Clauza GROUP BY**
- **4.3.2. Excluderea grupurilor (clauza HAVING)**
- **4.3.3. Imbricarea funcţiilor de grup**

copyright@www.adrian.runceanu.ro

4.3.1. Clauza GROUP BY

- Până acum toate funcţiile de grup au fost aplicate întregii tabele.
- Pentru a putea împărţi tabela în grupuri mai mici se foloseşte clauza **GROUP BY**.
- Folosirea acesteia returnează informaţii sumare despre fiecare grup.

copyright@www.adrian.runceanu.ro

4.3.1. Clauza GROUP BY

- Folosind **GROUP BY** nu se pot extrage şi coloane individuale, ci doar coloane ce rămân identice în tot grupul.
- Folosind **WHERE** se pot exclude rânduri, înaintea împărţirii lor în grupuri.
- Nu pot fi folosite aliasuri de coloane în clauza **GROUP BY**.
- Implicit, rândurile sunt sortate crescător după coloana (coloanele) specificate în **GROUP BY**.
- Acest lucru poate fi schimbat folosind **ORDER BY**.

copyright@www.adrian.runceanu.ro

4.3.1. Clauza GROUP BY

Exemplul 8 - Salariul mediu pe fiecare department:

SELECT deptno, **AVG**(sal)
FROM EMP
GROUP BY deptno;

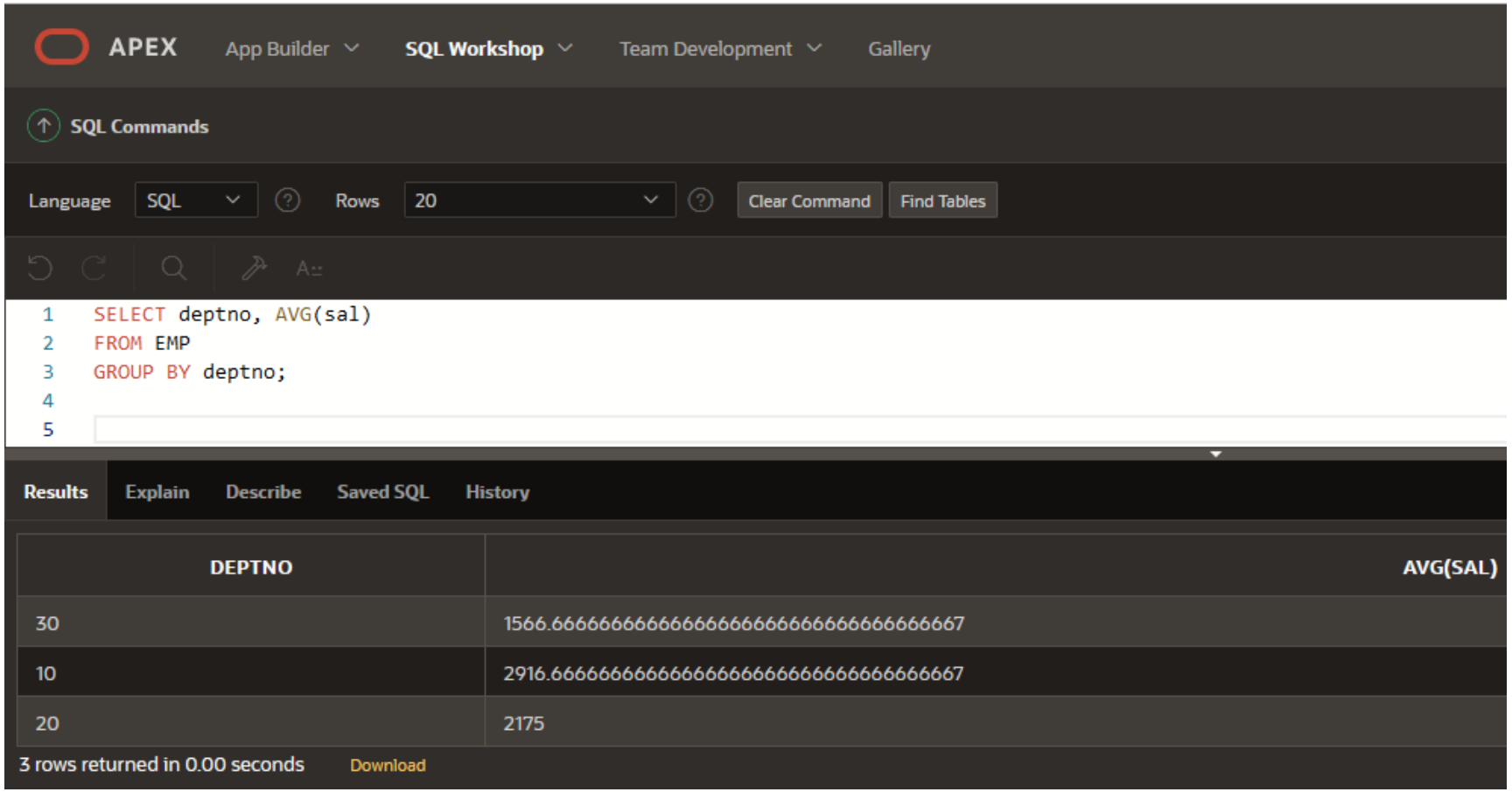
Column Name	Data Type	Nullable	Default	Primary Key
EMPNO	NUMBER(4,0)	No		1
ENAME	VARCHAR2(50)	Yes		
JOB	VARCHAR2(50)	Yes		
MGR	NUMBER(4,0)	Yes		
HIREDATE	DATE	Yes		
SAL	NUMBER(7,2)	Yes		
COMM	NUMBER(7,2)	Yes		
DEPTNO	NUMBER(2,0)	Yes		

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		11/17/1981	5000		10
7698	BLAKE	MANAGER	7839	5/1/1981	2850		30
7782	CLARK	MANAGER	7839	6/9/1981	2450		10
7566	JONES	MANAGER	7839	4/2/1981	2975		20
7788	SCOTT	ANALYST	7566	12/9/1982	3000		20
7902	FORD	ANALYST	7566	12/3/1981	3000		20
7369	SMITH	CLERK	7902	12/17/1980	800		20
7499	ALLEN	SALESMAN	7698	2/20/1981	1600	300	30
7521	WARD	SALESMAN	7698	2/22/1981	1250	500	30
7654	MARTIN	SALESMAN	7698	9/28/1981	1250	1400	30
7844	TURNER	SALESMAN	7698	9/8/1981	1500	0	30
7876	ADAMS	CLERK	7788	1/12/1983	1100		20
7900	JAMES	CLERK	7698	12/3/1981	950		30
7934	MILLER	CLERK	7782	1/23/1982	1300		10

copyright@www.adrian.runceanu.ro

4.3.1. Clauza GROUP BY

Rezultatul obtinut - Salariul mediu pe fiecare department:



copyright@www.adrian.runceanu.ro

4.3.1. Clauza GROUP BY

Exemplul 9

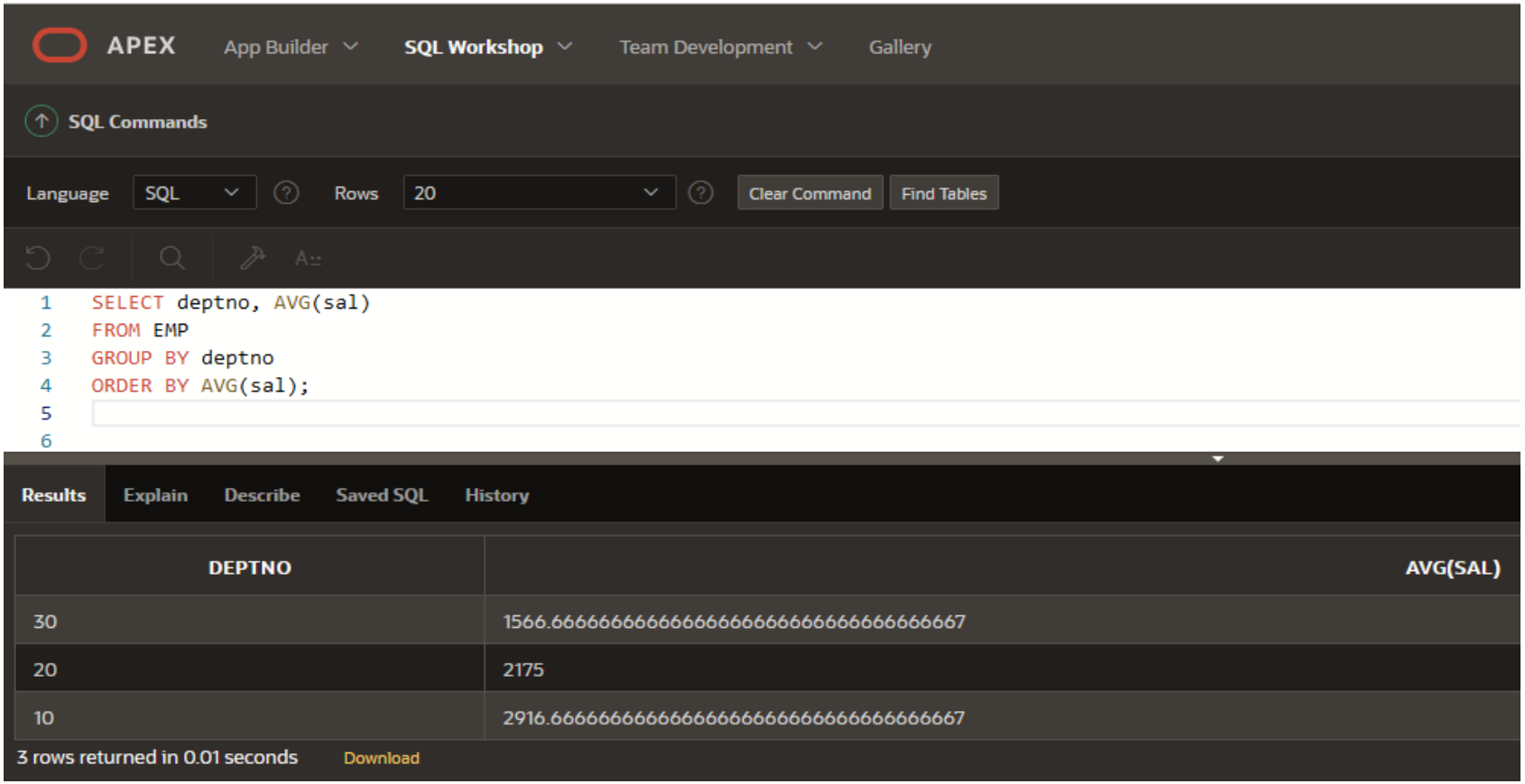
Salariul mediu pe fiecare departament, iar rezultatele ordonate după salariul mediu pe departament.

```
SELECT deptno, AVG(sal)
FROM EMP
GROUP BY deptno
ORDER BY AVG(sal);
```

copyright@www.adrian.runceanu.ro

4.3.1. Clauza GROUP BY

Rezultatul obtinut - Salariul mediu pe fiecare departament, iar rezultatele ordonate după salariul mediu pe departament.



copyright@www.adrian.runceanu.ro

4.3.1. Clauza GROUP BY

Gruparea după mai multe coloane.

Câteodată este necesară obţinerea de rezultate pentru grupuri în alte grupuri.

Atunci în dreptul clauzei **GROUP BY** vom întâlni mai multe coloane.

copyright@www.adrian.runceanu.ro

4.3.1. Clauza GROUP BY

Exemplul 10

Salariul total pe fiecare departament si pe fiecare functie, iar rezultatele ordonate după salariul mediu pe departament.

```
SELECT deptno, job, SUM(sal)
FROM EMP
GROUP BY deptno, job
ORDER BY AVG(sal);
```

copyright@www.adrian.runceanu.ro

4.3.1. Clauza GROUP BY

Rezultatul obtinut - Salariul total pe fiecare departament si pe fiecare functie, iar rezultatele ordonate după salariul mediu pe departament.

SQL Commands — Schema

Language: SQL — Rows: 20 — Clear Command — Find Tables

```
1  SELECT deptno, job, SUM(sal)
2  FROM EMP
3  GROUP BY deptno, job
4  ORDER BY AVG(sal);
5
```

Results | Explain | Describe | Saved SQL | History

DEPTNO	JOB	SUM(SAL)
20	CLERK	1900
30	CLERK	950
10	CLERK	1300
30	SALESMAN	5600
10	MANAGER	2450
30	MANAGER	2850
20	MANAGER	2975
20	ANALYST	6000
10	PRESIDENT	5000

9 rows returned in 0.03 seconds — Download

copyright@www.adrian.runceanu.ro

Limbajul SQL

Cereri SELECT pe o tabela

4.1. Funcţii

4.2. Funcţii referitoare la o singură înregistrare

4.3. Funcţii referitoare la mai multe înregistrări

- **4.3.1. Clauza GROUP BY**
- **4.3.2. Excluderea grupurilor (clauza HAVING)**
- **4.3.3. Imbricarea funcţiilor de grup**

copyright@www.adrian.runceanu.ro

4.3.2. Excluderea grupurilor (clauza HAVING)

Clauza **HAVING** funcţionează în mare ca şi clauza **WHERE**, diferenţa fiind că **HAVING** este folosit pentru a exclude anumite grupuri din rezultat, nu rânduri cum făcea **WHERE**.

Clauza **HAVING** poate fi folosită înainte de **GROUP BY**, însă este mai logic să fie folosită după.

Ordinea execuţiei va rămâne aceeaşi.

copyright@www.adrian.runceanu.ro

4.3.2. Excluderea grupurilor (clauza HAVING)

Exemplul 11

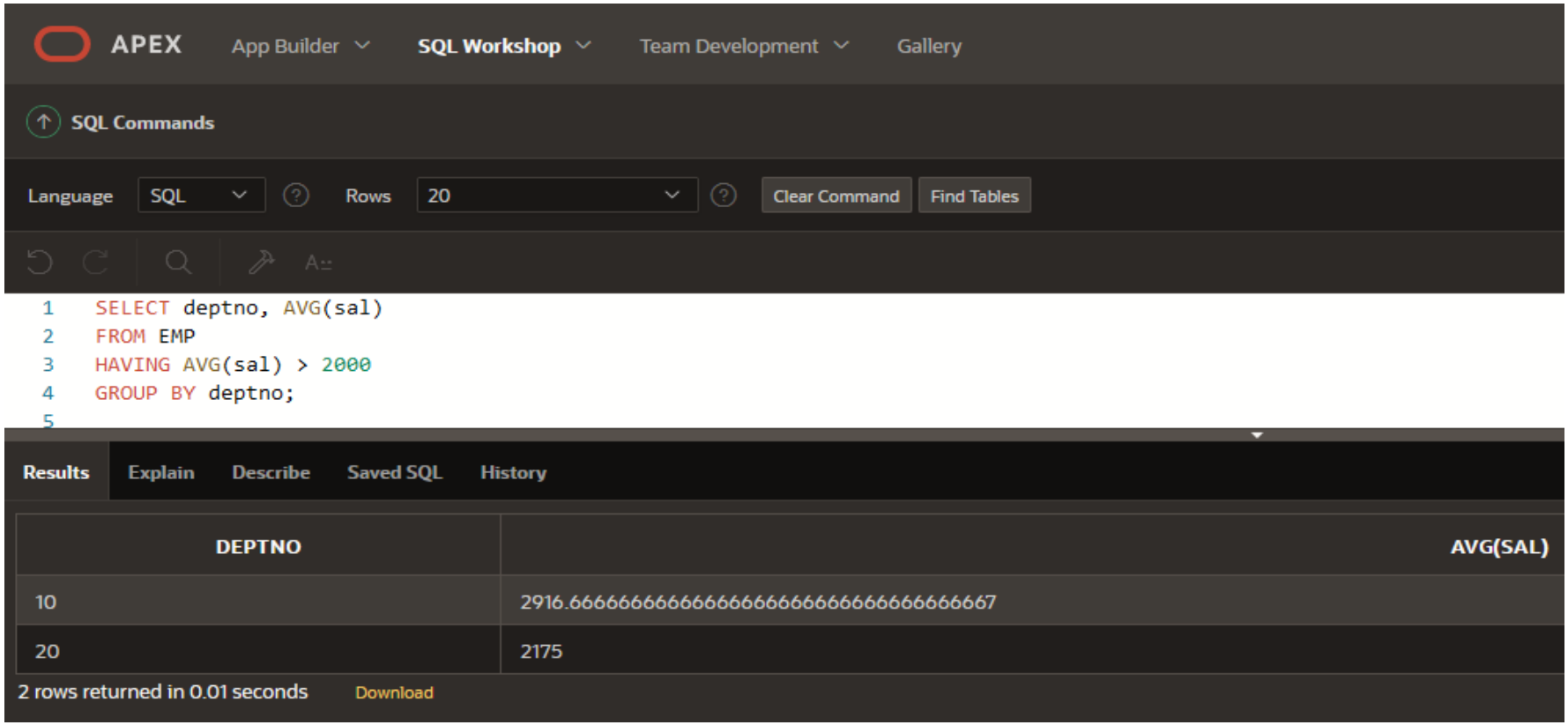
Salariul mediu pe fiecare departament unde acesta depăşeşte 2000$.

```
SELECT deptno, AVG(sal)
FROM EMP
HAVING AVG(sal) > 2000
GROUP BY deptno;
```

copyright@www.adrian.runceanu.ro

4.3.2. Excluderea grupurilor (clauza HAVING)

Rezultatul obtinut - Salariul mediu pe fiecare departament unde acesta depăşeşte 2000$.



copyright@www.adrian.runceanu.ro

4.3.2. Excluderea grupurilor (clauza HAVING)

Exemplul 12

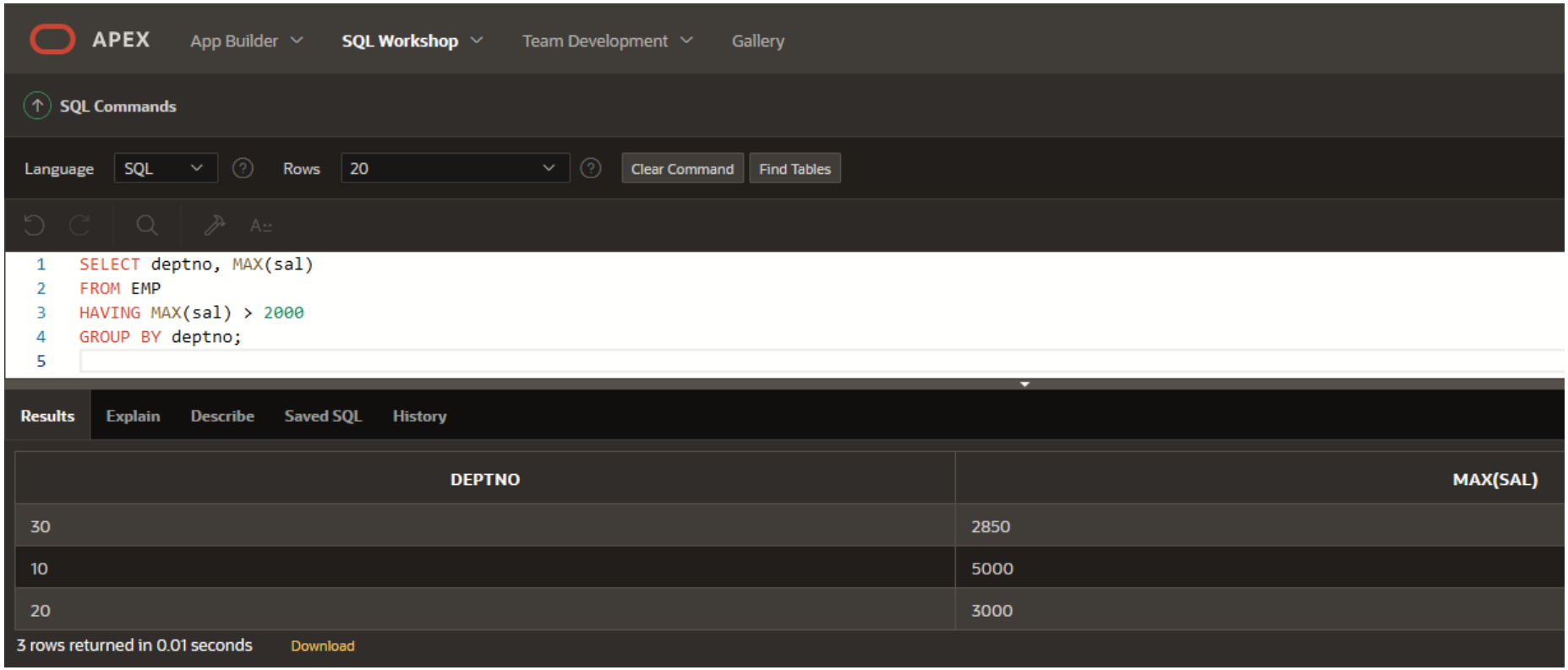
Salariul maxim pe fiecare departament unde acesta depăşeşte 2000$.

```
SELECT deptno, MAX(sal)
FROM EMP
HAVING MAX(sal) > 2000
GROUP BY deptno;
```

copyright@www.adrian.runceanu.ro

4.3.2. Excluderea grupurilor (clauza HAVING)

Rezultatul obtinut - Salariul maxim pe fiecare departament unde acesta depăşeşte 2000$.



copyright@www.adrian.runceanu.ro

4.3.2. Excluderea grupurilor (clauza HAVING)

Exemplul 13

Salariul total pe fiecare funcţie, fără a lua în calcul MANAGERII, excluzând funcţiile cu suma salariilor sub 6000$ cu ordonare după total.

```
SELECT job, SUM(sal)
FROM EMP
WHERE job != 'MANAGER'
GROUP BY job
HAVING SUM(sal) < 6000
ORDER BY SUM(sal);
```

Salariul total pe fiecare funcţie

fără a lua în calcul MANAGERII

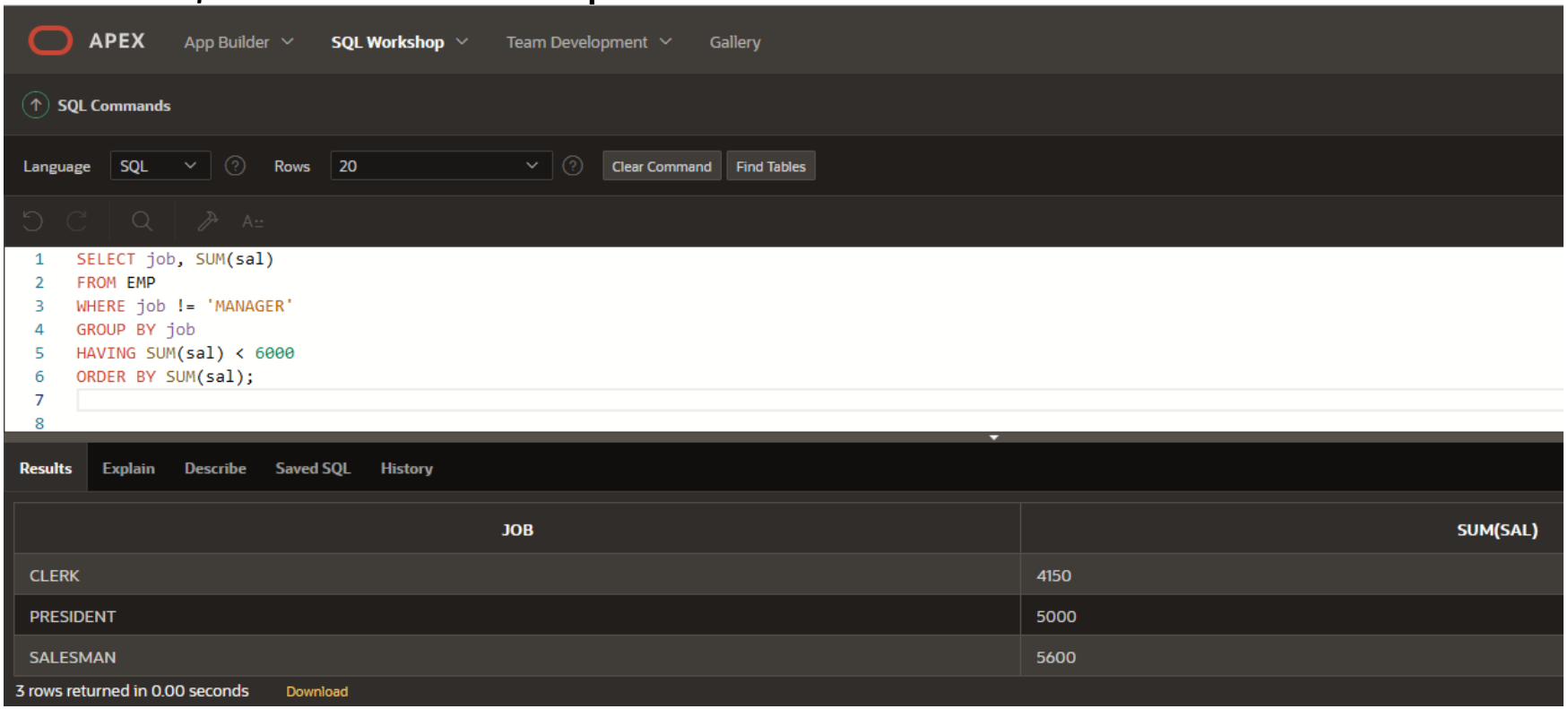
excluzând funcţiile cu suma salariilor sub 6000$

cu ordonare după total

copyright@www.adrian.runceanu.ro

4.3.2. Excluderea grupurilor (clauza HAVING)

Rezultatul obtinut - Salariul total pe fiecare funcţie, fără a lua în calcul MANAGERII, excluzând funcţiile cu suma salariilor sub 6000$ cu ordonare după total.



copyright@www.adrian.runceanu.ro

Limbajul SQL

Cereri SELECT pe o tabela

4.1. Funcţii

4.2. Funcţii referitoare la o singură înregistrare

4.3. Funcţii referitoare la mai multe înregistrări

- **4.3.1. Clauza GROUP BY**
- **4.3.2. Excluderea grupurilor (clauza HAVING)**
- **4.3.3. Imbricarea funcţiilor de grup**

copyright@www.adrian.runceanu.ro

Ordinea de executie a functiilor de grup

Serverul **Oracle** execută funcţiile de grup într-o anumită ordine:

1. Selecţia rândurilor ce respectă clauza **WHERE**
2. Gruparea rândurilor obţinute, respectând clauza **GROUP BY**
3. Calcularea rezultatelor funcţiilor de grup pentru fiecare grup în parte
4. Eliminarea grupurilor ce nu respectă clauza **HAVING**
5. Ordonarea rezultatelor respectând clauza **GROUP BY**.

copyright@www.adrian.runceanu.ro

Ordinea de executie a functiilor de grup

- Ordinea de execuţie are o importanţă foarte mare, deoarece are un impact direct asupra vitezei.
- Cu cât mai multe înregistrări pot fi eliminate utilizând clauza **WHERE**, cu atât mai puţin va dura gruparea şi operaţiile ce urmează.
- Dacă o cerere **SQL** este concepută să elimine înregistrări/grupuri doar folosind clauza **HAVING**, atunci ar fi bine de încercat dacă este posibil şi prin clauza **WHERE**. De obicei, totuşi, această rescriere nu va fi posibilă.

copyright@www.adrian.runceanu.ro

4.3.3. Imbricarea functiilor de grup

Funcţiile de grup pot fi imbricate cu o adâncime de 2.

Exemplul 14

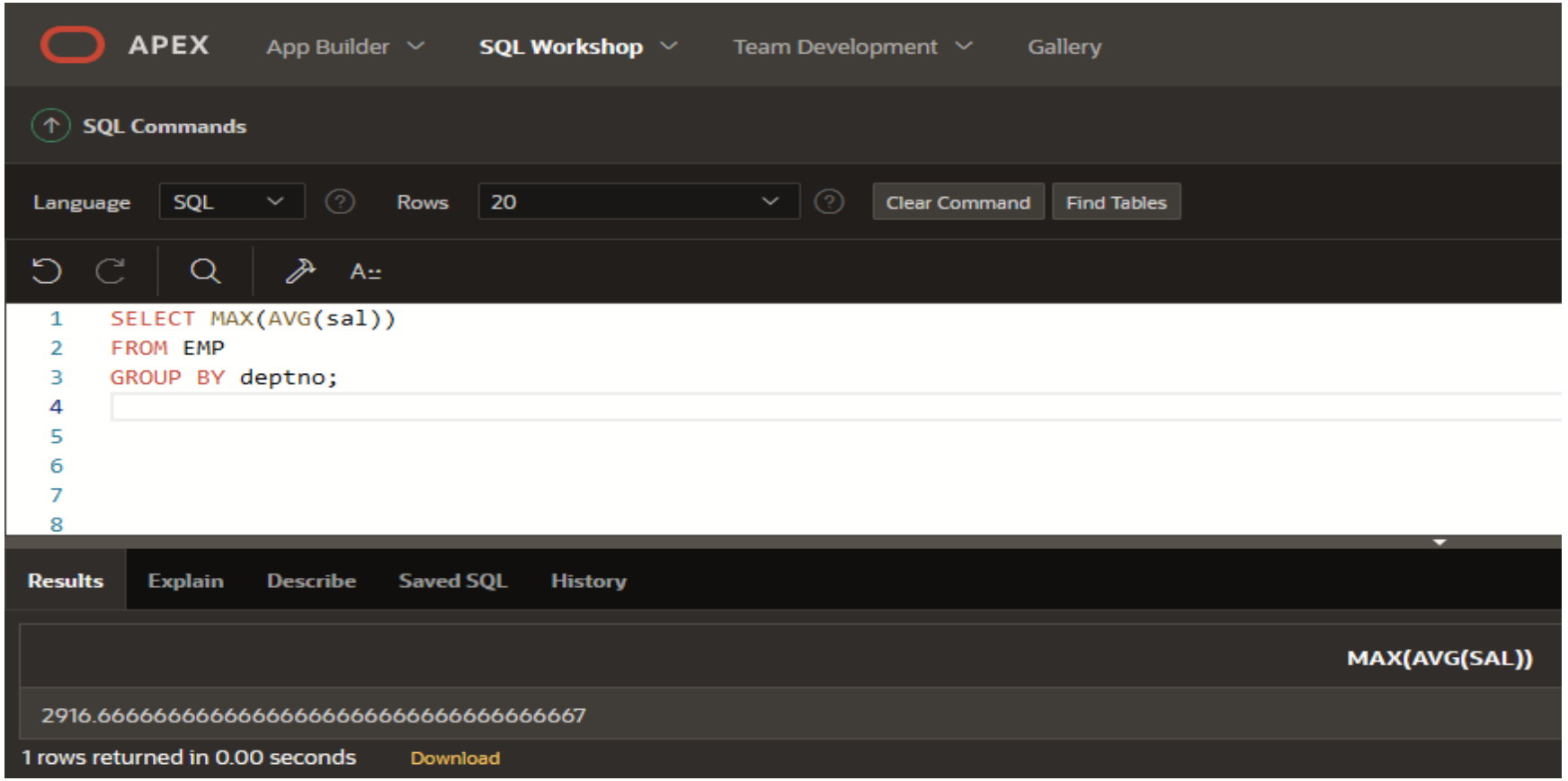
Salariul mediu maxim.

```
SELECT MAX(AVG(sal))
FROM EMP
GROUP BY deptno;
```

copyright@www.adrian.runceanu.ro

4.3.3. Imbricarea functiilor de grup

Funcţiile de grup pot fi imbricate cu o adâncime de 2.
Rezultatul obtinut - Salariul mediu maxim.



copyright@www.adrian.runceanu.ro

Întrebări?